A Study of Automotive Security - CAN Bus Intrusion detection Systems, Attack Surface, and Regulations

Master of Science in Technology Thesis
University of Turku
Department of Computing
Security of Networked Systems
July 2022
Alberto Giust

Supervisors:
Jouni Isoaho
Akwasi Adu-Kyere

UNIVERSITY OF TURKU
Department of Computing

Alberto Giust: A Study of Automotive Security - CAN Bus Intrusion detection Systems, Attack Surface, and Regulations

Master of Science in Technology Thesis, 62 p.
July 2022

The innovation in the automotive sector enhanced the technology implemented in vehicles by the manufacturers. Consequently, the overall driving experience improved, thanks to the introduction of better safety, utility, and entertainment systems. Moreover, automobiles began collecting and exchanging data with the external world through different communication protocols. However, these additions have started to attract attention from security experts. More importantly, malevolent attackers have exploited the technologies and their related attack points to carry out malicious activities to cause data security and safety issues. These issues have led to establishing standards and regulations (ISO 21434, UNECE 155, etc.) that redefine vehicle design and development by incorporating security protocols and requirements necessary to create secure automobiles. However, these documents analyze the problem at a high level and do not dwell on practical solutions implementation analysis. This work presents an in-depth study of in-vehicle communication concerns via Controller Area Network (CAN) bus safety problems analysis with different proposed solutions. Specifically, a survey of Intrusion Detection Systems developed in the literature is brought up: simulation of three CAN bus intrusion detection systems against various attacks. The results show effectiveness against disruptive attacks, i.e., with numerous messages sent in a short period of time, but conversely have difficulty detecting more targeted attacks with few transmitted packets. The solutions analysis is an excellent starting point for security engineers to be able to develop Intrusion Detection Systems for the CAN bus capable of detecting attacks that will become increasingly complex and difficult to counter over time.

Keywords: Automotive, Cybersecurity, CANbus, Intrusion Detection Systems, Security, ISO/SAE, Security Regulations, Security Solutions, Work Study

Contents

List of Figures

List of Tables

1 Introduction

In recent years, technological innovation in the automotive sector introduced new features in vehicles, quickly becoming systems connected to the outside world. Implementing technologies like Bluetooth, Tyre Pressure Monitoring Systems (TPMS), WiFi, and cellular communication changed the way vehicles are imagined, from merely mechanical devices to advanced electronic products. The ability to contact remote servers or other vehicles has improved the driver and passenger experience, simplifying several processes that were considered tedious in the past. These essential changes bring to light several vehicle security and privacy issues. In the past, the only way to attack a vehicle was to physically access the cockpit to unlock the components for tempering. Now, this is no longer the case. As vehicles communicate with the outside world, attackers begin to exploit the large attack surface[1][2][3] to gain unauthorized access to internal services, stored data, and communications between Electronic Control Units (ECU). Many systems developed do not factor in security and privacy in mind. As a result, many carried out attacks cause significant problems for manufacturers. Security experts begin to analyze the problem [4] and discover several issues that the industry must solve in the near future especially concerning the design and development of systems that follow security procedures. The results of these studies are several standards and documents [4][5][6][7] that in the near future will be introduced in the automotive sector to introduce the concept of security in all phases of the development of new vehicles. These regulations will become fundamental for manufacturers because conformance would be required for newer vehicles and will cover the entire development process, from design to sales and after-sales monitoring.

1.1 Aims and objectives

The first goal of this thesis is to study different CAN bus intrusion detection systems, simulate them and compare the results against different attacks to define what techniques are more effective in detecting anomalies while the vehicle is operating. The study describes the data used, the source, its modifications in order to inject the attacks, the CAN IDS used for the case study, the implementation procedure, the software used, and how the results were collected. The second objective is to present the state of the art of automotive security attack surface. The study focuses on how the attack surface expanded after introducing new vehicular functionalities and how these additions impact security, privacy, and concerning attacks in the automotive industry. The third objective is to analyze automotive security regulations in the upcoming future, their goals, how they will impact the industry, and compliance requirements.

1.2 Scope

The scope of this thesis is in two areas, the first of which includes the study and simulation of three anomaly-based CAN bus intrusion detection systems [8][9][10] against denial of service, fuzzing, and message replication attacks. The data is from a regular traffic recording of a heavy-Duty Truck(HDT) provided by the University of Turku. Malicious packets are injected (following different criteria explained in the thesis) into those recordings to simulate the attacks. The based analysis of security regulations comprehends automotive security standards released by the International Organization of Standardization (ISO)[1], other regional regulations, and best practices published in Europe[7] and America[6].

1.3 Structure

The structure of this thesis is below. Chapter 2 contains a state-of-the-art analysis of automotive security, the most recent attacks, and the security regulations that will come into force in the following years. The content of this chapter follows the research work accomplished during an internship conducted in Yarix[2], where all the main security regulations have been analyzed and compared. The company's objective is to have a complete study on this sector, new to them, to understand if it is interesting to invest in it. The recent release of ISO/SAE 21434 [4] is the starting point, but the research expands to other regulations and a more practical view of vehicle attacks and attack surfaces. Chapter 3 details CAN bus security, its limitations, and the most suitable security solutions, with a depth study on intrusion detection systems. Chapter 4 consists of a practical analysis of CAN bus intrusion detection systems developed in the literature, comparing their effectiveness against different attacks. The results are then discussed and compared. Finally, Chapter 5 brings the conclusions of this project.

[1]`https://www.iso.org/home.html`

[2]https://www.yarix.com/en/

2 Literature Review

2.1 Vehicle Security State of the Art

Vehicles evolved substantially during the last few decades. From complete mechanical and analog systems, they became digital products, until today, where they can be considered computers, "computers on wheels"[11]. The main objective of this technology shift is to develop products with better performance, but it is not the only one. It is also crucial for simplifying and improving the driving experience and obtaining information useful to find the destination more efficiently, communicate with the external world, gain information about the system's status, and create a safer environment for the driver and the passengers. The introduction of the ECU in vehicles in the 1980's changed the design of vehicles completely but, at the same time, raised concerns about the security of these systems. The safety of drivers and passengers was always the first concern for automotive manufacturers. On the other hand, introducing electronic components, but more importantly, opening the cars to the outside world using different communication protocols, made OEMs rethink their primary goal. Shifting from a closed to an open system requires putting safety and security on the same level of importance. In IT systems, a vulnerability not fixed signifies a loss of Confidentiality, Integrity, or Availability, while in IoT or the automotive sector, the consequences can be more serious (think of an attacker with complete control of the brakes of the vehicle).

2.1.1 Attack Surface

Nowadays, a vehicle has up to 150 Electronic Control Units[12], embedded computers that control different aspects of electronic components, manage sensor data, and send information between each other. Different networks and network protocols communicate information through a car or a truck, which would be secure if the whole system were closed from the external world. Unfortunately, there exist many different ways for the external world (people, devices, other networks) to interact with a vehicle, and Figure 2.1 gives an overview of it.

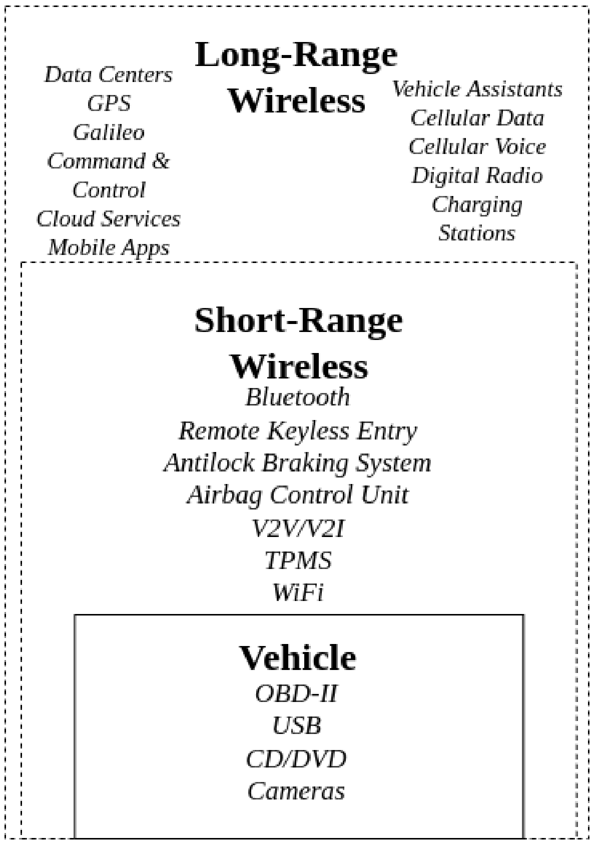

Figure 2.1: Vehicle Attack Surface

The vehicle attack surface can be divided into different areas, depending on the distance from the vehicle.

Vehicle physical access gives the opportunity to directly access the On-Board

Diagnostic (OBD-II) port, a diagnostic port installed as a mandatory interface since the 1990s in the United States and 2003 in all the vehicles in EU[13]. Its initial introduction was to facilitate mechanics' work to diagnose possible faulty or broken parts. Standard ISO 15031[14] defines how to communicate between a vehicle and a diagnostic interface external to it.

OBD-II port directly connects to the different communication buses, like the Communication Area Network (CAN) bus, or lower speed network like Local Interconnected Network (LIN). Connecting to these networks allows an attacker to get information about the vehicle's state. Moreover, they can inject data into it, tampering with the original flow with malicious actions. Nevertheless, attacks on the OBD-II port are more challenging to implement since an attacker needs to be in complete physical control of the vehicle. This port can communicate with an external interface if is turned on, so keys are necessary most of the time. A slightly more feasible scenario would be to tamper with the interface or the device connected to the vehicle for diagnostic (e.g. a laptop) to gain remote control of the car.

A vehicle often has other kinds of ports that can be interacted with and that are commonly used for entertainment [15]. Firstly, CD/DVD players are becoming less popular these days, but they are still present in most of cars. A poor configured player or a vulnerability in the reading process can lead to code injections in the entertainment unit of the vehicle[1]. An attacker may be able to insert some spyware or even plant ransomware to block part of the entire car's functions completely. A similar scenario may happen with USB ports and USB devices [16], which are often blindly trusted and can cause serious harm to the system. USB devices are one attack vector, but it is essential to remember that smartphones are now playing a massive role in interacting with cars. Generally speaking, we commonly name

[1]`https://www.theregister.com/2016/01/26/hackers_can_take_full_control_of_car_os/`

indirect physical access all those methods that require a physical connection to the car but do not require the attacker to be on-site to launch the attack. When they instead have complete control of the vehicle, and they can also modify the mechanical parts of it, we call it direct physical access.

Moving on to remote access, we can consider two other categories: short-range wireless access and long-range wireless access. Nowadays, they are the most exciting point of action for an attacker. Those features present a serious threat to what are now considered connected vehicles. These complex and intelligent systems interact with the outside world to provide a better experience and simplify short and long trips.

Speaking of short-range wireless technologies, Bluetooth is one of them. Users usually connect their smartphone with the vehicle to receive calls, control the entertainment system, or even use it as a key fob[2]. Bluetooth uses a confirmation mechanism to verify the other end of the connection, but in some cases, connections are trusted. This situation opens the possibility of a various range of attacks.

All premium cars have a WiFi network installed inside the vehicle. It is an excellent addition for entertainment during the trip for passengers but it is also beneficial for transferring data from the vicinity of the vehicle, like software or firmware updates. This technology is widely used in IT systems, and is widely tested, but misconfiguration or lack of hardening can lead to vulnerabilities.

Another interesting technology widely implemented is the Tyre Pressure Monitoring System (TPMS), which is used to continuously get the status of the tires and alert the driver of low pressures. It became mandatory in Europe in 2012[17], and is beneficial for real-life monitoring Unfortunately, the communication is poorly secured because there is no encryption, and the transceiver module supports a sig-

[2]`https://www.tesla.com/ownersmanual/model3/en_us/GUID-E004FAB7-1C71-448F-9492-CACF301304D2.html`

nificant range, meaning that it is easier for black hat hackers to craft a successful attack. Sensors, in general are targets for attacks because they are used to ask the central system (for example, sensor ECUs that codify the data read and send messages to highly critical ECUs) to take immediate action. Radar, laser, or LiDAR are some examples of sensors that can be tricked by attackers to act maliciously. Moreover, the world of autonomous vehicles is even more critical because they use cameras for object and sign detection and artificial intelligence to interpret data to take the right action. ENISA recently published a study [18] that explains how crucial these technologies are for autonomous vehicles. They are fundamental for the majority of actions, and at the same time, dependent on the outside world. Modifying a street sign can trick artificial intelligence into producing wrong results. Moreover, human eyes cannot see some of these anomalies, so it is even more difficult to investigate them.

Finally, another emerging technology used in connected vehicles is Vehicle-To-Everything (V2X) communication. It comprises all the methods for a vehicle to communicate with the outside world, short-distance: other vehicles (Vehicle To Vehicle or V2V), infrastructures like signs, bridges, traffic lights (Vehicle To Infrastructure or VTI), or even pedestrians and bicycles (Vehicle To Pedestrians or VTP). This technology is new and not standardized yet, but it will significantly improve the driving experience thanks to faster and more precise information about dangerous situations or anomalous scenarios. On the other side, it raises security and privacy concerns because data from different sources are read and processed and needs to be stored. It will require anonymization and security measures to avoid tampering. At the same time, specific authorities require traceability since the participants of the networks need a way to verify that the other end is trusted[19]. These technologies are still developing, and they will become more relevant when autonomous vehicles are the majority because they will be the ones that benefit the most but security is a concern.

Long-range wireless communication is another crucial component in modern connected vehicles. One example is GPS, used to track vehicle location, or Galileo GPS navigation system[3]. Having a poorly secured sensor can cause a leak of information and can be used by an attacker for further malicious actions[20].

Analog radio is also an old technology that is slowly being replaced by Digital Audio Broadcast (DAB radio). It is also part of the attack surface because it is a way of communicating with the outside world and the vehicle. Radio stations sending bogus signals can be a severe threat if the vehicle is not adequately secured.

Finally, another technology widely used in vehicles is the cellular network that allows the car or the truck to connect the internet directly. It greatly benefits the experience of the passengers, and it can also be used for remote updates on the fly, entertainment, or for exchanging status information between backend services of the manufacturer. On the other side, it is the main access point for malicious actors that aim at getting remote control of the vehicle or stealing sensitive data. Global System for Mobile Communication (GSM) technology is already widely used and tested, but cannot be considered 100% secure.

2.1.2 Security Attacks

As shown in the section above, the attack surface is vast and covers different technologies. Subsequently, the idea of having non-hackable cars is far from possible, and literature has proven that manufacturers and suppliers need to take this problem seriously. In this section, an overview of the most exciting attack studies is presented in order to show what are the present state of automotive security and what are the weak points.

[3]`https://www.euspa.europa.eu/newsroom/news/first-galileo-enabled-autonomous-vehicle-successfully-demonstrated`

The older security attacks focused on physical access, primarily via the OBD-II port, generally used for diagnostic. Miller and Valasek[21] focused on the analysis of the packets, reversing the communication messages in order to inject instruction to target various ECU of two modern cars. These types of studies are a fundamental building block for more advanced attacks because in-vehicle networks are used to send critical information, and by accessing that, it is possible to craft dangerous attacks. Checkoway et al. [22] identified different vulnerabilities on a media player that allowed a CD formatted following a particular technique to be run and interpreted and a buffer overflow in its firmware that opened for code execution capabilities. Christensen and Dannberg[23] managed to perform a man-in-the-middle attack on an OBD-II diagnostic interface that allowed them to intercept all the data accessible via a cloud interface.

Researchers also focused on remote communication protocols and discovered various anomalies: Rouf et al.[2] analyzed two popular Tyre Pressure Monitoring Systems and found that they were vulnerable to eavesdropping and packet injection, and they managed to raise an alert on the system. This attack is not critical per se but allows malicious actors to induce the victim to stop and check the car.

Another category of attacks targeted the key fob used to open and close the vehicles. Samy KamKar's study[24] revealed how it was possible to record the signal sent from the key to the car and replay it to unlock it, thanks to the rolling code mechanisms that use this technology. The principle used was jamming the communication to avoid the signal reaching the vehicle and invalidating the following code sent to open the car.[3] presents a more advanced attack, where the Bluetooth module inside the key is exploited to perform a firmware upgrade and bypass the security controls to be able to open and turn on the cars. These attacks do not target moving vehicles to steal information about the driver or the location but instead aim at stealing the complete vehicle, which causes significant financial damage to the victim and the manufacturers.

In the history of vehicle security attacks, the study by Miller and Valasek on a Jeep Cherokee[1] was fundamental because it showed that attackers could tamper with cars remotely using newly integrated WiFi capabilities. This attack exploited a vulnerability in the entertainment unit of the car to access the wireless network of the car and then obtain remote code execution thanks to an open port. From there, accessing the CAN network was the final step to controlling highly critical systems, like acceleration, steering, and brakes. The consequences were enormous: FCA Crysler had to retire millions of vehicles because there was no possibility to upgrade the systems remotely in a fast and reliable way[4].

A recent study from Upstream security[25] gave an in-depth analysis of recent trends in security attacks in 2020 and revealed that the most targeted entry points are backend servers, key fob, and mobile apps. It clearly shows that the majority of attacks are caused and facilitated by the interaction between IT systems (more mature in terms of infrastructure and development life-cycle but at the same time, hackers have better knowledge of the technologies used). It emerges that connected and autonomous cars are complex systems that use different technologies, more or less advanced and protected: this situation may open numerous doors for the attacker to leverage unknown vulnerabilities. The development of a secure system is mandatory nowadays, and the sector has already taken action to provide a methodology to reach these goals in the immediate future.

The future of automotive will include the massive introduction of electric and autonomous vehicles in the market. These technologies have already proven how impactful they can be in society, but at the same time, they raise a security concern. Schneider Electric[26][27][28][29] discovered multiple vulnerabilities in electric charg-

[4]`https://www.theguardian.com/business/2015/jul/24/fiat-chrysler-recall-jeep-hacking`

ing stations that allowed remote execution and privilege escalation on the software beneath. Moreover, autonomous vehicles have numerous sensors that help detect the environment, and they use this data to make decisions like turning the steering wheel, braking, or turning the lights on. Petit and colleagues[30] showed how it was possible to attack the cameras and LiDAR of an autonomous car using relatively cheap hardware. The attack consisted in blinding the camera sensor preventing it from detecting obstacles using bright light sources. Meng et al.[31] studied the security of GPS systems in autonomous vehicles and proposed a GPS spoofing generator that modifies the original signal to a bogus one. The consequences of these attacks depend on the reaction protocol of the car, but they can cause serious harm to the passengers.

As shown by these attacks, security in the automotive sector is becoming crucial. Manufacturers must take action to secure the vehicle from the upcoming attacks, and manufacturers need to be the first to bring this to the attention of all the other companies. Since security was not considered an essential factor until these recent years, engineers build different technologies without any protection measures. The result is that attacks are easily achievable with cheap hardware. Moreover, the introduction of security measures as reactive tools without modifying the overall architecture is not always a viable solution: the introduction of security standards will help manufacturers in achieving this objective of implementing a new developing life-cycle.

2.2 Vehicle Security Regulations

The rising amount of cybersecurity attacks in the automotive sector should be considered an alarm that naturally leads to taking proper action. Introducing a security system is mandatory in every vehicle and every communication protocol, access point, and critical component. However, it requires a deep analysis of what the manufacturer currently has and proceeds carefully in changing all the processes to create a cyber security management system. Here is where security standards can help. The International Organization for Standardization (ISO)[5], for example, aims to develop standards for every aspect of life in order to improve the quality and the security of products and processes. One of the most critical standards in automotive is ISO 26262 "Road vehicles - Functional safety"[32] which focuses on presenting a framework that helps vehicle companies develop safer vehicles. These aspects are fundamental because cars are products that directly interact with the users, and a malfunction can cause serious harm. On the other hand, it completely lacks a description of the security aspects of the vehicle and how to implement security.

2.2.1 Cybersecurity Regulations

In 2016, the Society of Automotive Engineers published the J3061 Cybersecurity guidebook for cyber-physical vehicle systems, the first standard that addresses the development of vehicles and vehicle components following cyber security principles. This document describes how to apply cybersecurity to the whole process of developing a vehicle, from start to finish and even after the selling. J3061 has been recently superseded by another standard, which starts from this document as a baseline and builds up a comprehensive analysis of cybersecurity in automotive. It is the ISO/SAE 21434 "Road vehicles - Cybersecurity engineering"[4].

ISO/SAE 21434

Standard ISO/SAE 21434 was published in August 2021, and its goals are:

- brings cybersecurity to all the actors and processes involved in developing vehicles and vehicle components

[5]`https://www.iso.org/home.html`

- create a common ground for implementation of cybersecurity processes and create a common vocabulary
- raise awareness across the industry

The document consists of different parts. Each part highlights fundamental aspects of security, how to introduce security and security culture, what the risks are, how to calculate the risks, and what manufacturers must consider when releasing new products.

More specifically, chapter 5 explains how to introduce cybersecurity inside an organization, who is responsible for instituting and maintaining a cybersecurity culture, and a risk management system following related standards ISO31000 [33]. Chapter 6 explains what are the requirements to fulfill during each new project, which is project-specific, like the cybersecurity plan, the reuse of existing components, and the inclusion of third-party resources. Chapter 7 lists all the cybersecurity activities not phase-dependent, for example, continuous monitoring, event assessment, vulnerability analysis, and vulnerability management. Chapter 8 details risk assessment methods used by companies to define what are the impacts of a threat they may face in the future, starting from the asset definition to the risk determination and treatment. There are also suggestions of practical tools: for example, STRIDE model [34] (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of privileges, which was initially developed for IT threats, but vehicle security can take advantage of that since they share some similarities in the infrastructure) for threat modeling, damage scenario assessment (with damage categories for Safety, Financial, Operational and Privacy), or attack feasibility rating with CVSS or attack potential approaches. Chapter 9 describes how cybersecurity is added on top of the concept phase, what the context is, how the context could influence security, and what the cybersecurity goals are. Here is introduced the concept of Cybersecurity Assurance Level (CAL), a classification model used to verify if a new component follows guidelines. Chapter 10, Product Development, describes how to define requirements from high to low level and how these requirements should be verified. The V-Model explains that this process is continuous and needs to be refined during all the development, adapting according to changes. Chapter 11 explains the process of cybersecurity validation at the vehicle level to verify that the initial claims are verified and the residual risk is acceptable. Chapter 12 covers production processes and what manufacturers must consider to avoid unnecessary errors or problems that can be difficult to resolve at this stage. Chapters 13 covers operations and maintenance. After releasing the product, it may be necessary to take immediate action to respond during a cybersecurity incident, so it is necessary to have a plan ready for use. Moreover, the security team could install an instant update to fix the weak point. This solution will probably be the first option in the future. However, it is still quite challenging to implement since vehicles must constantly be connected and they must trust the software downloaded over the air. Chapter 15 covers the decommissioning of the product. Finally, chapter 16 describes the interaction between the actors involved in the development (manufacturers, tier 1 and tier 2 suppliers), communication between them, and the process of taking responsibility.

As shown by this summary, ISO/SAE 21434 gives an overview of the development process in automotive and how to integrate cybersecurity, listing requirements, recommendations, and work products. Also, it brings an example of an actual component covering all the design phases. This document could be a great innovation, and manufacturers will probably start implement them in the following years. However, it does not provide details on fulfilling all the requirements. It leaves some sections to be implemented depending on the situation and context. For example, cybersecurity monitoring and activities used to gather information regarding emerging threats or possible incidents during post-production activities are not well defined. However, they will be fundamental in the era of connected cars. A recent analysis brought by Trend Micro [35] also explains other monitoring sources, external to the organization and tools directly installed in the vehicle, that can be useful in performing these activities. The incident response team could monitor the network traffic, looking for anomalies, bug bounty hunters can perform black-box testing on online and offline components, and the analysis of the underground market helps anticipate possible zero-day vulnerabilities.

Other regulations and best practices

ISO/SAE 21434 aims to become a standard in the automotive industry, and stakeholders will probably start to require it on all new products, but it is not the only one. Countries and organizations released numerous documents to help improve cybersecurity in the sector while keeping high-quality products. In 2020, United Nations Economic Commission for Europe (UNECE) released important regulations addressing cyber vehicle risks, incident response, and software updates[36] (they were later on approved in 2021 in [5]). More specifically, the first document details the definition of a cybersecurity management system with specifications regarding threats and vulnerabilities. ECE/TRANS/WP.29/2020/79[25]) can be summarized in these points:

- backend servers are a great source of data, and an unauthorized attacker may be able to access this information easily if the communication with the vehicle is blindly trusted
- in-vehicle communication systems are fast and reliable but not secure
- software updates are possible attack targets. Attackers can exploit this feature to install malicious software
- human factor is still one of the most critical assets, and it is not easy to secure.

Education is the only solution

- if in-vehicle communication is crucial, external communication opens numerous possibilities for attack
- software code is already one important asset in IT systems, and vehicle manufacturers must not ignore it and enforce controls to secure it as much as possible.

This regulation will affect more than 50 countries in Europe and worldwide. Starting in 2022, manufacturers must follow these rules when releasing new products. Starting from 2024, these regulations will affect every vehicle.

In 2016, National Highway Traffic Safety Administration (NHTSA) released "Cybersecurity Best Practices for the Safety of Modern Vehicles" [6], and they updated it in 2020. It is a collection of best practices that manufacturers should follow during design and development. It also contains a section on education, a fundamental aspect in this context: they suggest working closely with universities and research centers to have a better view of new products and discoveries in the security world.

Auto-ISAC also released a series of best practices[37] that helps companies in enhancing automotive cybersecurity, but they are not mandatory. ENISA good practices for security of Smart Cars [7] adopted a similar approach, focusing on automotive and semi-automotive vehicles (from automation level one to five). The document also describes asset and threat taxonomy, including examples of attack scenarios and how to prevent these incidents.

As shown by this overview, different sources released different documents. However, they aim at reaching similar results: achieving acceptable levels of security is fundamental, and OEM needs to take this problem seriously. Security by design could be a valuable summary of the whole content: it means that the process of building a new component for a vehicle puts security in the first place, and the design follows guidelines and recommendations based on a risk assessment adequately conducted. Cybersecurity strictly depends on and directly impacts safety because a weakness in the security of a system can cause catastrophic consequences for the safety of human users.

On the other side, these documents lack the practical side: requirements and recommendations give what should be the outcome but rarely explain the how (for example, ISO/SAE 21434 has some examples, but most guidelines are theoretical and leave the company to implement concretely).

2.2.2 Data Privacy Regulations

Data Privacy is as important as cyber security and strictly correlated to safety. Consequences of poor security could be data leaks, with a severe impact on user's privacy. Moreover, the automotive sector will deal with clients' data more often. Vehicles will produce terabytes of data per day[38] in the near future because some technologies require the usage of a vast amount of information. Backend servers store the data so that the vehicles can use this information to interact with the environment. Some security regulations tried to address data privacy problems, but other documents are specific requirements for this area.

In Europe, the principal regulation of data is the General Data Protection Regulation (GDPR)[39]: from May 2018, data collection and manipulation followed rigid rules enforced to protect the data owners from a leak, losses, or unintended usage. The sector of automotive needs to comply with it because the majority of data handled and transmitted contains sensitive information. Since every OEM handles data, it is considered a data controller[6]. So it is responsible for protecting the users' information, taking precautions that avoid possible data leaks, and directly informing the affected actors in case of incidents. GDPR promotes the concept of Privacy

[6]`https://upstream.auto/blog/gdpr/`

by Design. Like security, privacy must be considered the first goal of every new product. The development needs to follow guidelines and methodologies that help in reaching the requirements asked by the regulation.

The European Data Protection Board also released a document [40] listing essential guidelines to follow in collecting and managing data specifically for the automotive sector, following law directives. The authors explain how to manage data inside and outside the vehicle and what anonymization and pseudo-anonymization are. Moreover, they present some case studies showing how to apply these regulations.

3 CAN Bus Security

Controller Area Network is a bus protocol designed and developed by Bosch in 1986 [41] that became standard in the early 1990s with CAN Specification, version 2.0[42]. Nowadays, the CAN bus is one of the most used communication technology for in-vehicle communication, and it did not evolve much over the years. The principal objective of the CAN bus is to have a communication network and standard protocol to send and receive messages reliably (it is robust from electrical and electromagnetic interference) between the different ECUs installed in the car without high costs and weight. One of its main features is the usage of only two wires (CAN High and CAN Low)to communicate in all the vehicles. Each node in the network can send a message if it gains precedence (multimaster), and every node can receive the message sent (broadcast). CAN bus consists of the implementation of the data link layer and physical layer of the ISO/OSI model[43], described in ISO 11898[44].

CAN bus provides up to 1Mbit per second transmission speed, and it can go up to 5 Mbit per second in CAN FD, a newer generation of CAN with a flexible rate, developed and released in 2015. Today, every moving vehicle (and even advanced industrial machines) use CAN protocol. Some of them have multiple networks with different speed: low-speed CAN connects non-critical controls, like light, doors, or windows, while high-speed CAN connects the engine, transmission, brakes, suspension, ... Manufacturers can install gateways as mediators between the different subnetworks installed in the vehicles. These components may also serve as security controllers. If a node wants to transmit data, it is the transmitter, while the receiver is who accepts the messages (since the technology used is broadcast, a node can decide to accept or ignore a message based on the implementation). The message is sent using a differential voltage between the two cables, where one is represented as zero voltage differential (recessive bit), and zero is represented as five voltage differential (dominant bit). Carrier Sense Multiple Access with Bitwise Arbitration (CSMA/BA) controls how the nodes access the network. When multiple nodes want to send a message simultaneously, they send the arbitration ID, and the lower ID gets the bus and can start sending the entire message. Lower IDs represent more critical ECUs that have precedence compared to others (for example, brake signals).

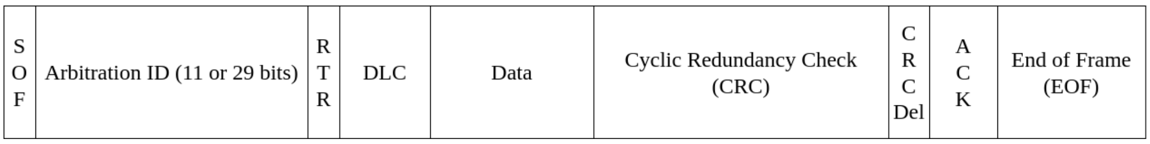

Figure 3.1: CAN bus schema

The proper name of CAN messages is CAN frames. They consists of different fields (see Figure 3.1):

- **Start Of Frame** (SOF). A single dominant bit.
- **message identifier** or **arbitration ID**. It is 11 bits or 29 bits long (depending on if the protocol follows CAN v2.0A or CAN v2.0B implementation), and it uniquely indicates the sender ECU, meaning that in a CAN bus, each node must have a different ID.
- **Remote Transmission Request** (RTR). It indicates if the message is a data message or a remote message.
- **Data Length Code** (DLC). It indicates the length in bytes of the data (0-8).

- **Data Field**. It contains the message's data, and the DLC field indicates its length.
- **Cyclic Redundancy Check** (CRC). The content of this field defines the integrity of the message.
- **CRC delimiter**
- **Acknowledge**. It asserts that the message arrived (set to 0)
- **ACK delimiter**
- **End Of Frame** (EOF). It indicates that the frame has ended, and every byte transmitted after that does not regard this transmission

A CAN frame always falls in one of the following types: data frame, remote frame, error frame, or overload frame.

Data Frame. It is the most common frame. It contains data that the transmitter wants to send to another node. Its structure consists of SOF, arbitration ID, DLC, data, CRC, ACK, and EOF.

Remote Frame. It has the same structure as a data frame, but without data. A node sends this frame to ask for some information from another ECU.

Error Frame. A node immediately transmits an error frame (6 dominant bits and an error flag) if it identifies an error. Every other node that receives this message automatically responds with another error frame. CAN bus also implements an called Error Confinement Mechanism (ECM) that disables nodes that send multiple error messages over time. Each node turns on in an Error Active state and they can send error messages. Every time an error frame is sent, there is a counter that increases. When the counter reaches 127, the node turns into an Error Passive node and it cannot send dominant error messages. After 255 messages, it turns off, completely stopping its functioning[45].

Overload Frame. A node uses it when the rate of the messages is too high to handle by a node, the structure is similar to an error frame, and its goal is to slow down the communication.

3.1 Security Challenges

CAN bus protocol was designed and developed to resolve the rising amount of connection needed between vehicle components caused by the usage of point-to-point protocols. The result was an efficient solution, resistant to hardware malfunctioning or interference with reasonable speed and transmission rate. It was not built with security in mind, so there were no immediate solutions to mitigate external threats. However, it was not a critical issue before the era of connected vehicles. When manufacturers began to install new features that opened the cars to the external world, security specialists started to raise concerns.

A security assessment of the CAN protocol reveals that it lacks all the fundamental security aspects reunited in the CIA triad: confidentiality, integrity, and availability.

Confidentiality means that the asset analyzed has mechanisms to protect, with encryption and authentication, the information exchanged, and an external unauthorized entity cannot access this data. CAN bus does not implement encryption since the data frame is relatively small, so it is possible to retrieve the content of each message easily if it is possible to access the network.

Integrity is enforced when the content of a message is not modified between the source and the destination. If the information has been compromised, it should be clear when analyzing the packet, and this piece of data should be considered invalid. CAN protocol implements an integrity mechanism with the CRC field to check if the data field has been corrupted during transmission. However, there is no way to verify that a message, with the correct CRC, was previously tampered with or sent by another source. A message with coherent CRC is considered valid, even if an attacker was the creator. The fact that there is no authentication mechanism makes it easier to access the network and become an active node that can send messages.

Availability is the property that ensures that data is always available when requested by an authorized entity. Unfortunately, the CAN bus protocol is vulnerable to Denial of Service (DoS) attacks because the arbitration algorithm always gives precedence to messages with low IDs. A compromised ECU can start sending messages with ID 0x00, and the bus will be saturated with bogus messages, preventing the other nodes from exchanging genuine content.

A security assessment on the CAN bus deeply depends on the assets involved. Unfortunately, data transmitted on CAN is critical, and in [46] it is divided into three main categories:

- private data refers to information related to the driver's behavior, location, habits, and access key to personal devices directly connected with the vehicle.
- safety-critical data comprehend all the information related to the vehicle itself directly affecting its status, like the speed, the brake signals, and the status of the lights. An attacker can use this information to analyze the state of the vehicle
- vehicle security data is every bit of information that specifies the ownership of it, like the cryptographic keys used to open the doors, turn on the engine, to access protected areas of the system. A malicious actor could use this information to open a vehicle without consent or even steal the vehicle.

The security issues that emerged from the analysis of the CAN bus may have immediate consequences on the privacy of the data because the vehicle is no more a closed system, and the information can leave the vehicle networks when used for communication with other systems and to be analyzed. CAN data is critical for privacy, and the process of handling it is crucial to comply with regulations.

The severity of an attack on the CAN bus depends on what permissions an attacker gains by exploiting a node connected to the network or bypassing the gateways installed[47]. A weak ECU attacker refers to a control unit that has been compromised but is not entirely controllable. It can only send a frame with a specific arbitration ID. If the message has a lower priority, the attack can be less powerful and less impactful. The second type of attacker is usually called a strong ECU attacker, which is completely compromised, and the controller gains complete control over the message creation, the transmitter, and the receiver. In this scenario, the attacker can send a message at a customized rate containing arbitrary data. Consequently, they may be able to control the whole network and influence how the other nodes interact with each other, for example, by triggering multiple error messages that cause the complete shutdown of the targeted ECU.

Besides eavesdropping, an attacker may be able to inject messages in different ways to obtain different results. The most accessible type of attack consists in filling the bus with frames having the highest priority ID. When the attack wants to target a specific action instead, the ID is specific, resulting in an impersonation attack that causes the victim ECU to read messages with different content than the one expected (that resembles the actual state of the vehicle component). Replay attacks follow a similar methodology: the exploited component immediately sends the exact content of a message to trigger an action. By turning on and off the light, the vehicle uses the same message. As a result, a replay attack causes the lights to remain in the initial state (when the driver tries to turn on the light, the attacker immediately replies to the message, and the lights ECU triggers two times). Another type of injection consists in sending frames with random IDs: a fuzzing attack exploits this technique and can be used to enumerate the network if the attacker does not know what the ID of each ECU is. The result may be the bus mapping that can be used to craft other types of attacks. The last resource is modifying the content of the frame and testing what the consequences on the vehicle state are.

3.2 Security Solutions

CAN bus protocol has been known for not being secure by default. Researchers and industry designed different security measures for prevention and reaction. The first solution analyzed consists of network segmentation. It is a technique widely used in the IT sector. Its goal is to create different subnetworks, separated by security gateways, that cannot communicate without a real-time traffic analysis. This technology allows separating the ECUs in confined environments that are more difficult to access in case a compromised component tries to inject malicious traffic. Less critical subnetworks can be allowed to communicate directly with the external world, while sensitive messages must pass through different layers of control before leaving the secure area. Studies on CAN bus segmentation implement different topologies: Kammerer et al.[48] proposed a start topology using one central router that mediates the communication between the different ECU subnetworks. In this scenario, the router is in charge of controlling the traffic. It tries to avoid or detect possible attacks using different techniques, like enforcing message direction, analyzing the traffic data for integrity checking, and implementing a network intrusion detection system. Results showed that this solution is resistant to various types of attacks, but, on the other side, it adds a single point of failure and complexity to the network. In [49], network segmentation is implemented differently. During normal traffic, the network is not segmented. When an attack is detected, the malicious node traffic is redirected through a secondary network. The intrusion detection system is set between the primary bus and the ECU, preventing subsequent injections. The problem encountered in this study was to detect which node was the compromised one since there is no way for origin verification in the CAN bus.

Network segmentation is a solution that helps enforce cybersecurity in the CAN bus, but every node in the subnetwork can eavesdrop on other messages without any restriction. Introducing encryption in this protocol could solve this problem, and different studies tried to prove that the frame can be secured during transmission. Encryption is a technique that utilizes cryptography for securing data during transmission, where only the source and destination can read the correct message, thanks to an encryption key exchanged at a previous moment or already possessed. Some solutions aim at providing authentication in the bus: when a message is received, the destination is assured that the content of that frame has been sent by a trusted and known ECU.

Lu et al. [50] proposed lightweight and efficient encryption and authentication algorithm called LEAP. It consists of a key management system that uses pre-shared keys during development. These cryptographic structures are the starting point to produce a key that is different for each node, and the keys update frequently. Encryption and authentication are based on the RC4 algorithm to generate cryptographically secure messages that can be read-only by the sender and the destination. The algorithm was proven efficient against different attacks, with relatively low overhead (11 bits per message, which is around 17% of the total payload). The only weak point is a compromised ECU that can send correct messages to the other nodes, but the eavesdrop on the whole network is impossible. In [51] they propose a similar solution, an encryption algorithm using symmetric keys that can be enabled and controlled using a tool called CANTrack. Jukl and Čupera [52] implemented the Tiny encryption algorithm [53] in the CAN bus, a Feistel-type cipher considered lightweight and fast, and they proved that implementation is possible. Park et al. [54] used Elliptic Curve Cryptography to develop a secure communication system for the CAN bus. The message data is encrypted and authenticated, and different attacks failed during testing, thanks to their security mechanism. Unfortunately, the implementation is incompatible with the standard version of the CAN bus, like for [55], where researchers added encryption and authentication thanks to an FPGA chip.

Authentication is another fundamental aspect to be considered, and it may be implemented together with encryption because encryption mainly targets the confidentiality of data, while authentication targets integrity and source identification. In [56],they proposed a multicast authentication solution using message authentication code (MAC), and in [57], they revisited and upgraded the same concepts for better performance. However, in both cases, they were not proven to be backward compatible. A solution to this problem was presented in [58], where CAN+ protocol enforces authentication[59], that leverage out-of-band transmission to send 15 bytes, a signature to prove that the message is trustful and at the same time preventing replay attacks thanks to a counter. On the other side, the nodes need to install preshared keys before starting the process. Wang and Sawhney developed a security framework and authentication mechanism called VeCure that uses trust groups to rank different ECUs based on the probability of being targets for attacks. Based on that, messages are sent as pairs, where the second one contains authentication data.

Based on the analysis above, some considerations could be made on these solutions and their feasibility. The addition of network segmentation surely helps in confining different areas of the bus, preventing attackers from gaining more control of critical components. However, at the same time, the new point of failure becomes the security gateways. Moreover, this solution adds complexity to the network. Encryption and authentication can be a solution for preventing eavesdropping and injection. Unfortunately, the architecture has some limitations that are difficult to overcome. ECUs do not have significant computational power, and the addition of security algorithms for encryption adds time and space complexity that in these systems cannot be ignored (time is crucial for critical responses, and space depends on the message's maximum dimension). Introducing new hardware solves this problem but creates a problem for backward complexity and deviates from the concept of standardization that the CAN protocol aimed at establishing. A more viable solution is targeting the traffic analysis during its normal functioning, switching from a preventive approach to a reactive approach, ignoring hardware limitations of the nodes, and producing valuable data. Intrusion detection systems can solve this task, and the following section will present a detailed overview of this solution.

3.3 CAN Bus Intrusion Detection Systems

Intrusion Detection Systems are systems that analyze the network, the rate of the messages, and the information contained to check for possible malicious activities and raise alerts. They are used mainly in IT and other security solutions, like firewalls, because they offer a complementary approach for detecting security attacks. Intrusion Detection Systems are considered reactive solutions compared to firewalls, which are preventive (they limit the traffic). Reactive solutions may be considered less effective than others but, in reality, they offer excellent protection against attacks that cannot be detected by preventive systems (for example, attacks that manage to bypass the security barrier presented by a firewall).

As emerged in the analysis above, in-vehicle communication using CAN bus protocol and the nodes which access the network have limited computational capabilities, so it is pretty challenging to implement complex calculations using the given resources without modifying the backbone structure and standard. Consequently, introducing of Intrusion Detection Systems is a valuable security solution that does not require great protocol modification or a significant amount of resources. Every IDS usually are composed of three main parts: the sensing module used to read the data from the network, the detection algorithm, and the reporting module. The first component depends on where the device is installed (in the network, in the

ECU, or as a gateway [60]), and its goal is to read raw CAN data, add the necessary information, like the timestamp, and prepare it for further analysis. The decision on where to install the intrusion detection system influences the information retrieved during run-time: if it is host-based, the analysis will mainly regard the single ECU and what it does receive, while as network-based, the information would be less in-depth, but a broader overview of the network itself.

The detection algorithm is the core component of the IDS: based on the data received, the algorithm verifies if an anomaly or an attack has taken place. The techniques used are different and depend on what technologies are used and the knowledge of the state of the vehicle. Some IDS focus their analysis on lower-level characteristics, like the analysis of the analog signal [61]. However, additional and specific hardware is required to obtain valuable results. Other techniques focus more on the data transmitted, the rate of the messages, and the various fields' content. Finally, the reporting module's role is to take any action if an anomaly is detected. One possibility is to raise an alert, a standard behavior in IT. However, in the automotive domain, some considerations need to be made: it is not trivial to select how are the target for these alarms, the driver and the vehicle in general, or the back-end systems and the manufacturers. An IDS may also trigger some immediate response. However, it is challenging to decide which action is more suitable in which situation (stopping the car, slowing down, taking control over the vehicle).

Studies on intrusion detection systems are divided into different categories, based on the different detection approaches signature-based, specification-based, anomaly-based, and hybrid-based.

3.3.1 Signature-Based IDS

Signature-based intrusion detection systems are built on a knowledge base set that describes previously known attacks using some signature that, when detected, triggers the alarm. These solutions are usually not challenging to develop, and the detection accuracy is high when speaking of known attacks because they are immediately recognized since they follow a pattern. On the other side, unknown attacks are almost impossible to detect with these IDS because they do not follow previously known methods. These systems are not optimal when used in production since the most critical attacks are not known.

The advantage of using signature-based detection depends on the fact that automotive networks do not change their configuration during their lifetime, unlike IT networks where new servers constantly join the network because of expansions or redesign purposes. Consequently, the creation of specifications is limited and does not grow exponentially.

In [62], the authors propose a signature-based approach that uses information about the frame and its context. The study focuses on a theoretical formalization of the finite state automaton, implemented in a simulated environment. Results show partial success: regular traffic may be detected as malicious, and some patterns are not recognized if some frames are not analyzed properly. Jin and Chung [63] presented a lightweight IDS that detects anomalies based on traffic and content-related information. For example, a table containing the average time interval is presented. If the message does not arrive in a determined time frame, the system raises an error. Excellent detection rates have been proved for drop and reply attacks, but tampering is still not detected efficiently.

3.3.2 Specification-Based IDS

Specification-based intrusion detection systems use a different technique for detecting anomalies in the CAN bus. They collect a series of specifications that the protocol should respect during normal behavior, like a specific range of values, specific values that are dependent on other fields, or other characteristics that are useful to define regular traffic. Based on this information, a specification-based IDS can detect known and unknown attacks that deviate from the expected functioning. The specifications are installed before the algorithm starts, so they are known and coded manually (it will not be the case for anomaly-based IDS).

Larson et al. [64] developed a specification-based detection approach based on different behavior analyzed in the CANopen application protocol, packet-wise and network-wise. The results show that the detection is efficient, but some situations need reconfiguration to avoid possible false positives and false negatives. SAID-uCANT [65] detects anomalies based on the timing of the messages based on a supervised-learning method.

3.3.3 Anomaly-Based IDS

Different from signature-based IDS, an anomaly-based IDS usually does not need previous knowledge statically defined during the development on what are the possible anomalies or what are the signs of an attack. The algorithms used focus on the behavior of the CAN bus to define what is considered regular traffic, based on the frequency of the packets, the content of the messages, or other features. The great innovation in this solution lies in this concept because an anomaly-based IDS does not consider only already seen patterns and attacks but tries to expand its detection surface to unknown types of attack. It is crucial to expand the detection capabilities to unknown attacks because this sector is relatively new. It would be impossible to stay up to date with all the discoveries and, as already stated before, updating the vehicle functionalities is not feasible nowadays in a fast and reliable way. Anomaly-based IDS usually separate their functioning into two main phases: a training phase used to define what is considered normal behavior for the CAN bus traffic, and the actual execution phase, where the algorithm compares real-time traffic with the information gathered before to detect possible anomalies.

Anomaly-Based IDS fall into different subcategories based on the technique and type of anomaly algorithm used for the detection. Frequency-based IDS starts with the assumption that CAN messages are sent over the bus with a set frequency. Each frame has a transmission interval that indicates the waiting time between one message and another. Based on this notion, it is possible to profile the regular traffic and retrieve this value for each arbitration ID. After this phase, the IDS can detect anomalous messages sent out-of-phase out of its usual sending window. An attacker with limited access to the network will not be able to send malicious messages without an in-depth study of the network and without the capability of sending messages at a specific rate known by the manufacturer and on the same phase as the other messages. Song and Kim's research[66] is based on this concept. In their implementation, the IDS profiles the network taking the message rates as a significant feature: when a frame arrives, the timestamp is compared with the previous one with the same arbitration ID. If the interval is lower than half of the last saved interval, the IDS raises an alert. Moreover, it can detect possible Denial of Service attacks if the frequency of arrival is exceptionally high. The results proved that this solution is efficient with single and multiple CAN ID injections, with a low false-positive rate. A similar solution is presented in [8], where the time interval is compared with the previous one in order to detect high-frequency messages and out-of-phase messages. In [67], a different approach is proposed: on arrival time, the frame is checked to have a known arbitration ID and the previous messages do not share the same identification. It is possible to detect Denial of Service both with known and unknown frames and detect a suspicious sequence of similar messages. Hoppe et al.[68] developed an intrusion detection system based on the frequency of the message and their content. In this case, a context to the frame is added, it is analyzed, and a decision is taken based on the data (the signal to turn off a signaling light).

Machine learning intrusion detection systems are anomaly-based detectors that use supervised and unsupervised methodologies to investigate different characteristics of the bus to define what is regular and what is malicious. This branch of study is promising and robust thanks to its correlation abilities between features that usually are not detected by humans. At the same time, it has different drawbacks: these methods require much computational power for training and detecting. Moreover, the correlations found are not always meaningful and can lead to many false positives or negatives.

Kang et al. [69] propose an Intrusion Detection System using a deep neural network trained with CAN data packets generated using a simulation tool, and it has been proven to be effective in detecting attacks directed to the Tyre Pressuring Monitoring System. In [70] a Generative Adversarial Nets Intrusion Detection System (GIDS) is proposed: they use both real and generated data to train two different discriminators that are used to detect both known and unknown attacks. Results proved that the model responded better when trained with randomly generated data compared to real data because real data resulted in high performance only when tested against known attacks. Taylor et al.[71] developed an IDS based on a long short-term memory recurrent neural network used to predict the sequence of CAN packets in the immediate future using raw data. Researchers obtained good results in detecting attacks that add, remove, change the order of the frames or modify CAN data. In [72] an IDS using a Bayesian network is developed and analyzed, using detailed data from different ECUs (steering, brake, RPM, ...), and it was tested against cyberattacks on a virtual vehicle that is traveling on urban roads. Tanksale designed an Intrusion detection system using support vector machines based on three different feature groups (engine, transmission, and temperature), and they achieved high accuracy results in all the experiments. Haas et al.[73] proposed a possible implementation of an artificial neural network that uses packet information to detect malicious ECUs. The study is theoretical and not much information can be retrieved, but the authors argue that it is an interesting starting point for an efficient detector.

Finally, other anomaly-based IDS are listed below: [47] an intrusion detection system based on the physical crystal quartz clock. More specifically, they fingerprinted the ECUs based on the clock stew (the offset between the clocks and the true clock), and they modeled the detector that predicts the linear accumulation of this parameter and, in case of irregularities, an alert is raised. In [74] a graph-based IDS is proposed, based on the arbitration IDs occurrence and the utilization of a page rank algorithm. Wang, Lu, and Qu proposed an entropy-based IDS[75]: they analyzed the bits' entropy of the CAN arbitration ID, and they defined a threshold value for each bit to identify possible attacks that highly modify this CAN field (like DoS attacks that try to get full priority on the CAN bus). The proposed approach resulted in high accuracy when the attack sends lots of packets but struggles in identifying attacks that do not greatly influence the entropy by sending packets with low frequency or by replacing regular messages thanks to a compromised ECU. A similar approach is presented in [76] and [77] with similar results. Marchetti and Stabili proposed an IDS[9] that analyzes the message sequence of ID during normal transmission and raises an alarm when it encounters a new transition. They assume that there is a pattern between the messages transmitted during a normal situation, thanks to the periodicity of the CAN bus. So, they did not expect to detect a transition with an unknown ID or between two frames that are usually not sent in sequence. Results are very promising when considering a similar status of the vehicle. However, the IDS needs multiple retraining in case of different states: when the vehicle is moving, some messages are sent more frequently, while others are not sent at all, compared to the situation when the same vehicle is parked along the road. In [10], the hamming distance between consecutive CAN frames is used to detect anomalies, but instead of considering the arbitration ID, the payload is the discriminator. For each arbitration ID, the content of the consecutive frames is used to calculate this value and, if the values are too different from each other, based on training phase values, it raises. Lee et al. [78] designed an intrusion detection system that uses remote frames and response time to detect malicious actions.

Anomaly-Based intrusion detection systems are the most researched area of study thanks to their ability to detect attacks without the requirements of some signatures that identify the malicious actions. Moreover, the abundance of detection techniques, from frequency to machine learning, helps the researchers in exploring them and in developing new and improved versions.

3.3.4 Hybrid-based IDS

Hybrid-based Intrusion Detection Systems, as the name suggests, aims to merge different detection techniques to obtain the most effective and efficient solution. A large number of attacks and the variety of them create a lot of variability inside an in-vehicle network and some techniques may be more effective in a context while lacking in others. Weber et al. [79] developed a detection solution for CAN bus using a specification-based approach in the first phase to avoid possible false positives and then use the results of these static checks as input for different machine learning algorithms. A similar approach was followed in [80] where specifications and machine learning methods are applied to detect malicious attacks in CAN bus and automotive Ethernet. In [81], a set of rules is used to define the traffic considered normal, and a random forest model is used to detect five different attacks on a publicly sold vehicle.

4 Simulation of CAN Bus Intrusion Detection Systems

Previous chapters extensively discussed the importance of security and privacy in the automotive industry. The sector is trying to evolve at a fast pace, thanks to the introduction of regulations and best practices, and shortly, manufacturers will develop new vehicles with these aspects in mind. The introduction of security and privacy by default is a big step. However, they cannot be considered bulletproof, especially in this constantly evolving sector, where new functionalities are released regularly, and the possibility of missing something is increased. Monitoring and analyzing network traffic can be the solution for constantly controlling the vehicle for possible attacks.

4.1 Threat Model

The attack simulation starts from regular CAN bus traffic provided by the University of Turku (see Table 4.1), and new messages are added to simulate different attack scenarios. Studied attacks are the base of these scenarios described in the previous chapter.

timestamp	arbitration_id	extended	remote	error	dlc	data							
2021-06-22 14:11:21.676327	0x10fe6fe8				8	255	255	0	0	0	125	0	0
2021-06-22 14:11:21.676394	0x10f007e8				8	0	0	0	0	0	0	0	0
2021-06-22 14:11:21.677531	0x18fe5be8				8	84	0	0	0	1	8	0	0
2021-06-22 14:11:21.678995	0x10ff80e6				8	0	80	17	110	240	144	255	255
2021-06-22 14:11:21.679112	0x18f009e6				8	155	125	96	127	125	183	126	255
2021-06-22 14:11:21.680086	0x18fe59e6				8	242	124	255	255	86	126	76	126
2021-06-22 14:11:21.680197	0x18fec4b0				8	255	255	255	255	0	255	255	255
2021-06-22 14:11:21.683869	0xcf002e6				8	205	80	18	0	252	186	29	255
2021-06-22 14:11:21.683984	0xcf004e6				8	240	125	125	181	29	255	255	255

Table 4.1: Sample of clean data

4.1.1 Denial of Service attack

Denial of service attacks disrupts the normal behavior of a vehicle with multiple messages that usually have a lower ID because the attacker wants to get precedence during the arbitration phase and completely take control over the bus. This attack was developed by injecting CAN frames with ID 0x0, the lowest value possible, in intervals of 0.5, 0.1, 0.01 and 0.001 seconds (see Table 4.2).

timestamp	arbitration_id	extended	remote	error	dlc	data							
2021-06-22 14:11:21.676327	0x10fe6fe8				8	255	255	0	0	0	125	0	0
2021-06-22 14:11:21.676394	0x10f007e8				8	0	0	0	0	0	0	0	0
2021-06-22 14:11:21.677327	0x0				8	240	89	155	164	214	58	181	14
2021-06-22 14:11:21.677531	0x18fe5be8				8	84	0	0	0	1	8	0	0
2021-06-22 14:11:21.678327	0x0				8	142	169	121	213	150	121	126	48
2021-06-22 14:11:21.678995	0x10ff80e6				8	0	80	17	110	240	144	255	255
2021-06-22 14:11:21.679112	0x18f009e6				8	155	125	96	127	125	183	126	255
2021-06-22 14:11:21.679327	0x0				8	111	220	49	241	5	71	184	85
2021-06-22 14:11:21.680086	0x18fe59e6				8	242	124	255	255	86	126	76	126

Table 4.2: Sample of denial of service with injected message interval of 0.001

4.1.2 Fuzzing attack

Fuzzing attacks consist in sending random messages with random ID and data payload to trigger disruptive behavior during the normal execution of the protocol. Unlike the denial of service attack, fuzzing procedures do not aim at taking over the bus: instead, they aim at creating possible valid messages that are understood and parsed by the victim ECU. This type of attack was generated using two sets of ID: the first used a random set of ID, while the second used a valid ID taken from the specification of the Renault truck. The main difference between the two attacks is the amount of knowledge of the system by an external attacker: if they do not know anything about the system while the victim's ECU allows sending messages with a different ID, it is possible to send packets with random ID to guess possible combinations. On the other hand, if the attacker knows the specification of the CAN bus, they can send packets with valid ID and actively target the communication. Injected packets are sent each 0.5, 0.1, 0.01 and 0.001 seconds (see Table 4.3).

timestamp	arbitration_id	extended	remote	error	dlc	data							
2021-06-22 14:11:21.676327	0x10fe6fe8				8	255	255	0	0	0	125	0	0
2021-06-22 14:11:21.676394	0x10f007e8				8	0	0	0	0	0	0	0	0
2021-06-22 14:11:21.677327	0x18FEC4C8				8	192	198	88	191	252	91	5	40
2021-06-22 14:11:21.677531	0x18fe5be8				8	84	0	0	0	1	8	0	0
2021-06-22 14:11:21.678327	0x18E5FFA8				8	174	71	204	251	128	76	255	111
2021-06-22 14:11:21.678995	0x10ff80e6				8	0	80	17	110	240	144	255	255
2021-06-22 14:11:21.679112	0x18f009e6				8	155	125	96	127	125	183	126	255
2021-06-22 14:11:21.679327	0x0CFE6CE6				8	73	143	137	249	20	183	80	141
2021-06-22 14:11:21.680086	0x18fe59e6				8	242	124	255	255	86	126	76	126

Table 4.3: Sample of fuzzing attack with known ID with an interval of 0.001

4.1.3 Message replication attack

This attack does not want to saturate the bus nor randomly send messages with random ID: the main goal of this attack is to replay some CAN messages to replay an action that already happened, like turning the lights on/off or triggering the opposite action. The attack sends the transmission recorded a second time right after the first transmission occurred. More specifically, tests were made with a transmission length of 5.0, 10.0, 30.0, and 50.0 seconds, meaning, after the chosen time, all the traffic is replayed a second time, and the intrusion detection system will see all the packets in the same order.

The second method of replication consists in selecting a single message and replaying it after 0.1, 0.01, 0.001, and 0.0001 seconds, to simulate an advanced attack where a single action is performed by the attacker that has a better understanding of the system and obtained more privileges. The chosen ID is `0x18f009e6`, a message that is sent every 10ms during normal traffic transmission (see Table 4.4).

timestamp	arbitration_id	extended	remote	error	dlc	data							
2021-06-22 14:11:21.676327	0x10fe6fe8				8	255	255	0	0	0	125	0	0
2021-06-22 14:11:21.676394	0x10f007e8				8	0	0	0	0	0	0	0	0
2021-06-22 14:11:21.677531	0x18fe5be8				8	84	0	0	0	1	8	0	0
2021-06-22 14:11:21.678995	0x10ff80e6				8	0	80	17	110	240	144	255	255
2021-06-22 14:11:21.679112	0x18f009e6				8	155	125	96	127	125	183	126	255
2021-06-22 14:11:21.679212	0x18f009e6				8	155	125	96	127	125	183	126	255
2021-06-22 14:11:21.680086	0x18fe59e6				8	242	124	255	255	86	126	76	126
2021-06-22 14:11:21.680197	0x18fec4b0				8	255	255	255	255	0	255	255	255
2021-06-22 14:11:21.683869	0xcf002e6				8	205	80	18	0	252	186	29	255

Table 4.4: Sample of message replication attack log with interval of 0.0001

4.2 Intrusion Detection Systems

The study described in this thesis simulates three intrusion detection systems described in the literature. The author used the following factor to choose the algorithms: firstly, the papers need to have a clear and complete implementation of the IDS described, written as code, or pseudo-code, or other detailed instructions describing the parameter used. Secondly, the simplicity of the implementation is another important factor: as already stated multiple times, the detection must be fast, without massive overhead during the execution. Thirdly, the detection must be in real-time. When a packet or a series of packets are identified as suspicious, the intrusion detection system must be able to send the details immediately, without much delay caused by other calculations that need to be done after minutes of recording. Another critical factor is that the detection shall be possible with only software implementation, without any modification of hardware or protocol. Based on the type of simulation done, the intrusion detection systems are defined as network IDS because they monitor the traffic of every ECU that transmits using the CAN bus, so they expect to receive regular valid CAN messages during the simulation.

4.2.1 IDS based on the frequency of each message frame

The first IDS implemented was the one proposed by Gmiden et al. [8]. The main idea is to check each message time frame from each other (for example, when the IDS receives a message with ID `0xA1B`, it checks the timestamp of the last message received with the same ID). The IDS calculates the difference between the current time frame and checks if it is less than half of the shortest time frame calculated. In this case, it raises an alarm. Otherwise, if the time frame is just shorter, the new time frame is saved. The IDS ignores the remote frames (if the flag remote is set) and the answer to those (because we suppose that they are "out of regular frequency"). The detection based on the frequency starts from the assumption that CAN messages are sent periodically through the bus. If an ECU sends a message outside the expected period, the IDS must detect a possible attack. On the other hand, if an ECU is compromised and can send messages with its original frequency, the attack is not detected. Another disadvantage is the vehicle's state when the IDS is working: in fact, a vehicle sends some messages with different frequencies when it is running or when it is parked. If a manufacturer wants to implement this IDS, it must consider this factor. The solution can be a frequency-based IDS installed on the car that changes its mode according to the vehicle's state and saves the time-frames in different datasets to avoid unwanted collisions or false positives. In this study, we consider a single state of the vehicle the IDS is not connected physically to a CAN bus, but the IDS reads the messages from a log file, and there is no way to detect the different states. The IDS has no validation phase and can be run in real-time. When the IDS detects an anomaly, it immediately raises an alarm.

4.2.2 IDS based on the matrix of message ID transitions

The second IDS analyzed is developed by Marchetti and Stabili [9]. Its main assumption is that the messages flow through the CAN bus following a consistent pattern, meaning that it is possible to find sequences of message ID that repeat over time. In particular, during similar driving condition, a transition between `ID_1` and `ID_2` (first the ids receives `ID_1` and then the following message has `ID_2`) it is a consistent pattern. If the transition is new (at least one of the ID is new, or the pattern `ID_1 -> ID_2` is not in the database) an attack is probably happening. This IDS needs a training phase to build up a transition matrix within each row the origin ID and in each column the destination ID. It also needs a validation phase to verify that the matrix is complete and, if not, add more transitions. After that, the IDS is ready to detect in real-time if there are unknown transitions or unknown ID and raise alarms. One of the main strengths of this IDS is the ability to detect messages with unknown ID, but at the same time, it can be a weakness if the ID is valid but the training phase did not record it. In order to have a complete matrix, the training phase should accept a great number of messages.

4.2.3 IDS based on the hamming distance of frames with the same ID

The third IDS follows the idea of Stabili et al. [10]. Its functioning is based on the assumption that the content of CAN messages does not change much over time. The algorithm uses the Hamming distance to measure this value, which quantifies how much subsequent messages change. It calculates a range of valid values of hamming distance during the training and validation phase. During the detection phase, if a message has a Hamming distance smaller or larger than this range, the IDS raises an alarm because it is very likely that an attack is happening. Based on its definition, this IDS seems very good in detecting replay attacks and fuzzing attacks with malicious messages with an ID already seen. On the other hand, a message with an unknown ID is ignored, so attacks like DoS with high-priority messages can be ignored during the execution.

4.2.4 Implementation Details

The language of choice is python3.9 because it is a flexible language that can be pretty simple to write, but at the same time, it is very powerful and suitable for data manipulation and analysis.

Speaking of modules, the packet generation and manipulation happened with a module called python-can[82], a complete library for reading, writing, and analyzing CAN messages. It also allows to communicate with a real or virtual CAN bus directly, but during the simulation, this feature was not used since the messages were already recorded and saved into files. The design proposed by the library would have more overhead besides the original recorded time of arrival. Getting the log directly from a car and checking the validity of the IDS in real-time was the first idea. However, there was a need for expensive hardware and extensive study on a

	Predicted	
Actual	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Table 4.5: 2 x 2 confusion matrix that is used to classify the CAN messages during the analysis

chosen vehicle to tweak the bus reading accordingly.

The code developed for this simulation can be found at `https://github.com/alright21/CAN_IDS_simulation`.

4.3 Evaluation metrics

Some parameters and details were taken into consideration in order to define the efficiency of those systems. The study considered two metrics during all the tests. The first metric considered is the reaction time of the single intrusion detection system. As broadly described previously, this factor is crucial for deciding if the implementation is helpful in responding to attacks. If the response time is too long, the attackers may already have completed the attack when the system alerts the driver or the back-end network. On the other hand, if the detection does not have delays, the vehicle may be able to answer on time and prevent possible catastrophes. Using this metric is possible to compare how different intrusion detection systems react to different attacks and find out what is the most suitable in a specific situation.

The second metric is the detection rate of malicious packets during transmission. The main goal of this metric is to evaluate which messages are crafted by the attacker and which are not. High accuracy and precision help the designer of the security infrastructure to produce a trusted system that is helpful both in real-time during the attack, and for post-incident analysis, in order to define the causes, how the attack was implemented, and what are their particularities. The evaluation of this metric was calculated using the F1-score, a widely known scoring measure that considers precision and recall. These values are calculated by analyzing a 2x2 confusion matrix that identifies how the messages are classified by the intrusion detection system compared to the real evaluation of those messages. As shown in Table 4.5, each message prediction is compared to the actual value, and it assumes one of the following labels:

- **True Positive (TP)**: a malicious CAN message that is correctly identified
- **False Positive (FP)**: a legitimate CAN message that is identified as malicious
- **False Negative (FN)**: a malicious CAN message that is identified as legitimate
- **True Negative (TN)**: a legitimate CAN message that is correctly identified

Based on the evaluation of CAN messages it is possible to calculate the precision of the classifier (the intrusion detection system), that identify the rate between the malicious messages correctly identified over all the messages classified as part of the attack (see Equation 4.1).

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

Another important metric is referred as Recall, which indicates the rate of the malicious CAN messages correctly detected (see Equation 4.2).

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

Based on these two measures, it is possible to calculate the F1-score, which is a calculated combining these values and it represents the harmonic mean of the two[1] (see Equation 4.3

$$F1 - score = \frac{2(Precision * Recall)}{Precision + Recall} \tag{4.3}$$

[1]https://www.educative.io/edpresso/what-is-the-f1-score

4.4 Simulation

Simulation design and execution of the chosen intrusion detection systems use a simulation environment with a Lenovo X1 Carbon laptop (8GB of RAM), on a Linux Debian-based operating system.

Data preparation used python scripts for normalization. Every file consists of one-minute recordings: 60 files (one hour of recording) for training, and a single file (one minute) for testing, with injected packets. Data was read from CSV files, loaded in a simulated CAN bus sender, running on a separate thread, and sent through a CAN bus implemented using a python Queue. The choice of not using the python implementation of the CAN bus was straightforward: since data recorded had already propagation delay because it was recorded from a real CAN bus, the implementation didn't want to add additional overhead that was causing delay during the execution. The application schema is shown in Figure 4.1.

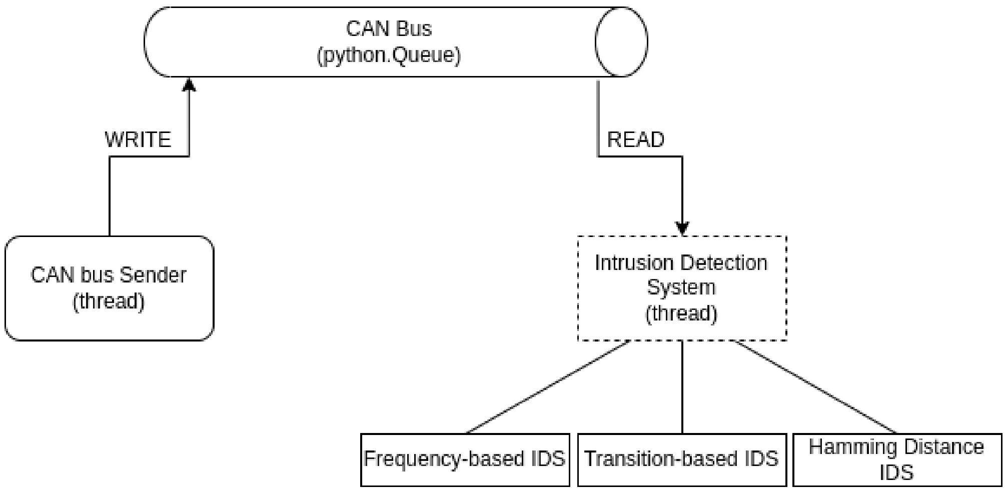

Figure 4.1: Schema of the simulation components

The simulation starts with a training part of one minute where the IDS reads regular CAN bus data and uses it to model their detector, based on the technique used. After this part, the attack starts, and the intrusion detection systems will be able to detect the attacks. Intrusion detection systems change their state to testing mode and start logging malicious packets. When the attack finish, a verification file is used to compare the results. By using this file, it is also possible to extract the time of the first detection.

4.4.1 Data preparation

Data used during this experiment was provided by the University of Turku: it was recorded from a Renault truck during standard driving situations. The data provided was raw, without any prior normalization or modification, and without specific details on the name of the truck, but it was enough for different practice tests. The communication was based on the SAE J1939 [83][84][85], a standard protocol stack used in heavy trucks and other work machinery. This protocol is used to implement higher-level communication, but the physical level is always based on the CAN bus protocol.

Data was provided in CSV format, and each row contained the following information (only the filled columns are described):

- **timestamp**: contains the packet's recording time. Since these files describe past screenshots of the truck CAN bus, this value is considered in the simulation as the packet's arrival time to the intrusion detection system.
- **arbitration_id**: contains the 29 bit extended ID of the packet.
- **dlc**: contains the length of the data field. This parameter can take an integer value from zero to eight since every CAN message has a maximum length of eight bytes.
- **data**: contains the actual data of the message, and it is represented in bytes, expressed as an integer with a value range from 0 to 255.

The process of message injection follows different steps based on the type of attack, the frequency of injection, and the type of message. Firstly, the file is loaded. Every attack starts immediately, so the first malicious message is printed in the output file. After this operation, actual messages and injected messages are alternated based on the type of attack. For example, in a denial of service attack, messages with ID 0x0 are injected by taking regular traffic data and analyzing the timestamp of the messages (when the logger detected the messages). The original file was given as input to a python program that analyzed each message. Based on the attack's behavior, the messages are injected (a new logged row was added). The final output file contains the original traffic and the new injected messages, respecting the correct timestamp.

4.4.2 Training Details

Every CAN bus intrusion detection system chosen for this simulation is defined as anomaly-based, meaning that the execution must be divided into two parts: training and testing. The training phase defines what is to be considered regular traffic without any injected packets. Intrusion detection systems use this information to build up the proper data structures that are used during the testing phase. Every IDS trains with the same data, 60 seconds of consecutive traffic recorded from the Renault truck. The actions taken by the simulated systems are the following:

- Frequency-based IDS analyzes the frequency of the packets with the same ID and stores the shortest legitimate delta time in a python dictionary.
- Transition-based IDS stores in a N x N matrix if it detects the transition between two ID during the training phase. The value of N is fixed if the number of valid ID is known (which is the case) or variable otherwise, but after the training phase, this number does not change because new transitions are flagged as malicious.
- Hamming-based IDS analyses the content of messages sent during the training

phase and stores a valid range in which the hamming distance is not considered suspicious and the IDS uses this information during the testing phase to detect possible malicious messages.

4.5 Results and Discussion

Figure 4.2 displays the performance results of the IDS against a denial of service attack: it is clear that Frequency-based IDS and Transition-based IDS performed flawlessly against this attack, mainly thanks to the choice of ID for the attack: they both didn't have experience of `0x0` messages. Consequently, alerts are immediately raised and every packet is correctly detected. On the other hand, Hamming-based IDS struggled a bit more because its detection method is not based on the ID but on the payload. The detection performance is not as high as the other two IDS. However, the majority of packets are correctly detected, especially when the attack is disruptive, and the frequency of injection is high since hamming distance becomes unpredictable fast. By adding information on the ID used during transmission in this intrusion detection system, it would be possible to detect all the messages correctly. In general, the performance of intrusion detection systems against denial of service attacks would change drastically if complete knowledge of the communication traffic is present during the design. When a message with an unknown ID is detected, it is highly probable that it is generated by a malicious source, and an alert is raised. This information would slightly change the classification of intrusion detection (also becoming specification-based IDS).

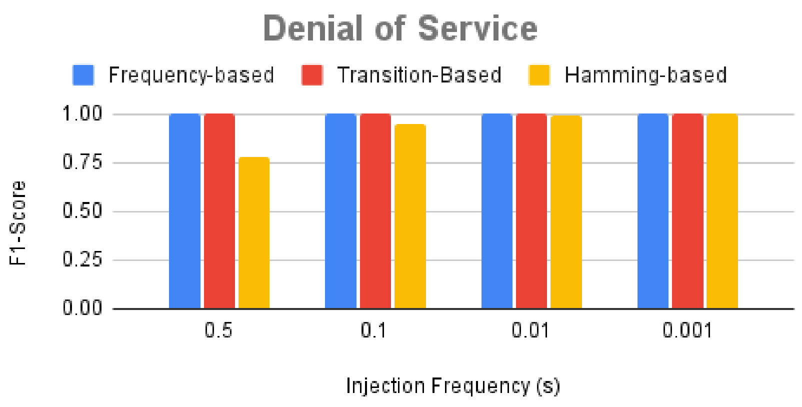

Figure 4.2: Denial of Service performance detection

Results against fuzzing attacks with random ID and payloads are displayed in Figure 4.3: the results are very similar compared to the previous attacks. Random ID are immediately detected by the first two IDS that raise an alert: F1-score is 1.0 for both, while for Hamming-based IDS the results are dependent on the payload, which is random. Since the goal of this attack is to find existing ID to send messages that are seen and processed by other ECUs, the success rate depends on the number of available ID and their length. Using extended ID drastically lowers the probability of actually finding a correct value. Consequently, this attack can be considered similar to a denial of service.

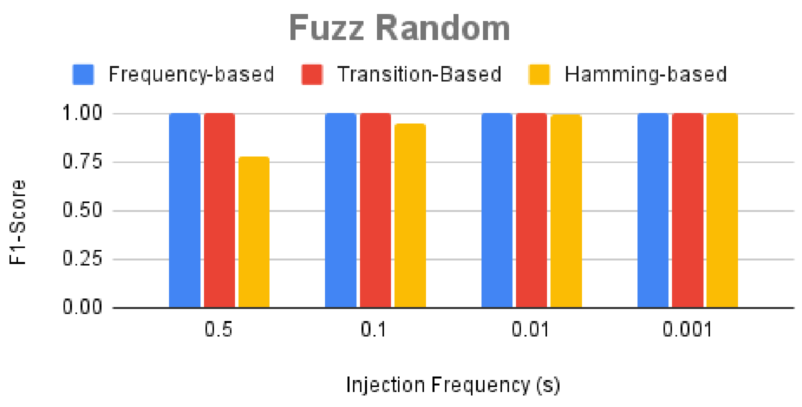

Figure 4.3: Fuzzing attack performance detection

Results obtain against fuzzing attacks with know ID are more interesting because the attack actively exploits knowledge of the system in order to send malicious packets to the CAN bus. As shown in Figure 4.4, F1-score obtained is on average 0.5, with Hamming-based IDS performing better than the others because random payload data is used during the attack and subsequently creates packets with a high distance between each other. Frequency-based IDS performed worst: the main idea of this IDS is to quickly detect high-frequency attacks, while targeted attacks usually remain undetected. Moreover, its success rate is also highly dependent on the training phase, where all the traffic is considered not malicious: if the CAN bus communication does not follow a frequency-based approach. However, often the message rate changes (e.g., a vehicle in different states sends messages at different rates), and detection can become unpredictable. Results of Transition-based IDS show better performance because this attack sends existing attacks without following a regular path. As a consequence, more invalid transitions are detected.

Speaking of detection time, all the IDS showed positive results, meaning that the first message of the attack is detected with no delay, meaning that an alert can be sent immediately (see Table 4.6).

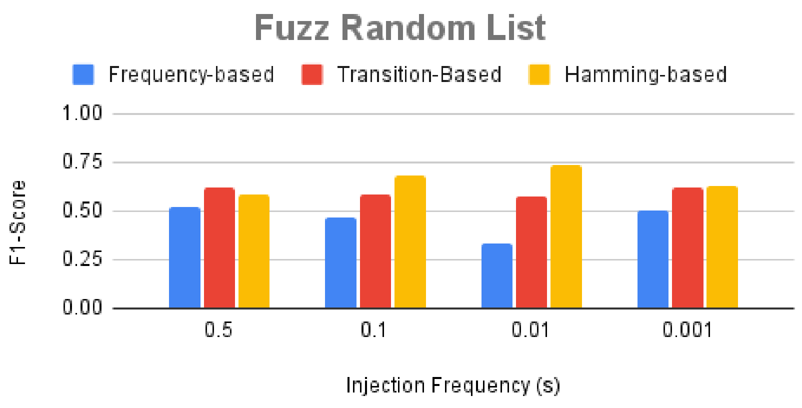

Figure 4.4: Fuzzing with known ID attack performance detection

Fuzz Random List	**0.5s**	**0.1s**	**0.01s**	**0.001s**
Frequency-based IDS	0.0	0.099	0.029	0.0
Transition-based IDS	0.0	0.0	0.0	0.0
Hamming-based IDS	0.0	0.0	0.0	0.0

Table 4.6: Fuzzing with known ID detection time

Replay attacks results are different: the first type of attack, where a portion of the traffic was replied to with the same frequency, and the same messages were not detected by any IDS. The causes are different for each of them: Frequency-based IDS could not detect any of the messages injected simply because the attack was so precise to perfectly mimic the actual traffic without any modification. Transition-based IDS struggled because the transition between the messages was previously seen and memorized during the training phase. As a result, the attacks were considered legitimate traffic, and no alerts have been raised. Finally, Hamming-based IDS was not able to detect the attack because the payloads were the same during the whole transmission, resulting in hamming distances already known and considered not malicious. However, this attack is theoretical: the implementation would require high knowledge of the system, complete control over the vehicle components, and the ability to stop legitimate traffic.

Finally, the last attack analyzed was the replay of messages with a single ID, with the intent of sending the same information a second time to cause an action opposite to the first one (e.g., turning the lights on or off). Figure 4.5 shows that Frequency-based IDS got good results, especially when messages are sent at a high rate. Transition-based IDS was mediocre in detecting the attack, both on the number of messages and detection time. Finally, Hamming-based IDS could not detect any malicious message simply because the content of the data frame was identical to the previous messages, resulting in no differences. Detection time are shown in Table

4.7.

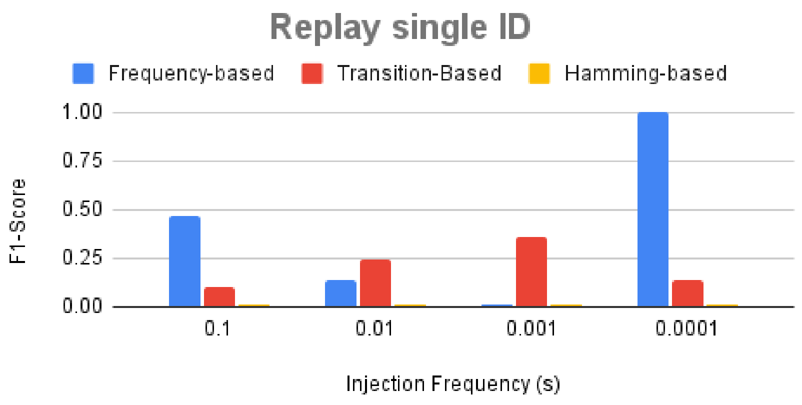

Figure 4.5: Replay single ID attack performance detection

Replay single ID	**0.1s**	**0.01s**	**0.001s**	**0.0001s**
Frequency-based IDS	0.0	0.05	X	0.0
Transition-based IDS	0.111	0.071	0.082	0.051
Hamming-based IDS	X	X	X	X

Table 4.7: Replay single ID detection time

Based on the results obtained during the simulation execution of these intrusion detection systems, it is possible to highlight some information gained, the positive and negative aspects, and define how these algorithms performed.

Firstly, it is essential to understand the environment where the tests have taken place, the hardware used, the language, how the IDS were implemented, and how the data was treated and used. Moreover, the attacks were explicitly generated for the purpose, without any accurate tests on the effectiveness of an actual vehicle, so it is pretty challenging to verify how this malicious traffic could affect the vehicle in a real scenario.

A positive aspect of the tests showed that every IDS successfully detected denial of service attacks with almost perfect scores, meaning that these solutions are effective against these types of disruption actions. By adding a Frequency-based IDS connected to the CAN bus, it would be possible to immediately alert the vehicle or the back-end systems that an attack is happening. Hamming-based IDS is highly effective when the payloads sent by the attackers are highly different from the original messages.

On the other hand, more advanced attacks, with fewer message rates and more knowledge on the attacking side, are more difficult to detect because the algorithms leverage high differences between a normal traffic situation, unknown ID, or random payloads. If an attacker started sending single messages targeting a particular ECU (with the correct frequency), it would be possible to bypass the detection. In this situation, if the payload of the messages were utterly inconsistent compared to the regular one, the Hamming-based IDS could be an excellent resource for detecting the intrusion. However, the success of this defense is dependent on how the payloads are generated, and what values they can assume in order to be valid.

The advantages and limitations of the intrusion detection systems compared revealed that frequency-based detectors are effective in detecting disruption attacks with a high volume of traffic. However, they are not effective in detecting targeted attacks with specific messages. Some improvements could be made to these systems to boost their performance. Firstly, it is suggested to add information about the CAN bus before the training start: for example, knowing the ID of the message exchange in a CAN bus can help to detect denial of service attacks. Moreover, switching the detection mode based on the state of the vehicle (idle, turning on, driving at a slower speed, driving at a higher speed) is another great addition that would help in differentiating how the training phase could improve the detection of more targeted attacks. One of the biggest problems encountered during the simulation was, in fact, that messages were sent with different frequencies and different payloads during the training phase, causing the thresholds used for the detection to be less strict. Multiple training phases would help the detection of single messages and, consequently a fast response.

As discussed during the presentation of this project, the simulation did not involve any machine learning-based IDS because the literature offers lots of theoretical studies but no simple solution that could have a practical implementation. More studies on this area are needed because it would greatly help in detecting some advanced attacks. Another improvement would be to combine different algorithms in order to obtain different alerts that can be cross-compared. Different IDS focus on different aspects of the traffic and different parts of the data frame. An intrusion detection system more focused on the message's ID could completely ignore an attack on the payload and vice-versa.

Finally, improvements could also be made in developing the code, how the IDS stores and analyze data, and how they handle the alerting. In the current implementation, all data is logged, and it is finally checked with the verification file to evaluate the performance. In a final deployment, the IDS should be able to immediately send an alert to a receiving system, which could be the vehicle or a back-end system located in a remote location (if the vehicle is connected to the cellular network). In recent years, the concept of the Vehicle Security Operations Center has become more interesting for vehicle manufacturers. A vehicle SOC consists of different technologies that, combined with in-person analysis, help find attacks on vehicle infrastructure. Different companies are already building these VSOC[2] [3] [4] [5] [6] [7] [8] [9] because the use of these systems will help manufacturers in respecting new regulations, on what concern post-development monitoring and incident analysis.

[2]https://argus-sec.com/argus-fleet-protection/

[3]https://upstream.auto/solutions/vehicle-security-operations-center/

[4]https://www.escrypt.com/en/solutions/vsoc

[5]https://cybellum.com/product-security-operation-center/

[6]https://www.autocarpro.in/news-national/denso-and-ntt-begin-validating-vsoc-tech-for-monitoring-vehicle-security-status-44916

[7]https://www.fujitsu.com/jp/solutions/business-technology/future-mobility-accelerator/mobility-security/en/

[8]https://news.panasonic.com/global/press/data/2021/03/en210323-2/en210323-2.html

[9]https://upstream.auto/resources/techtalk-capgemini/

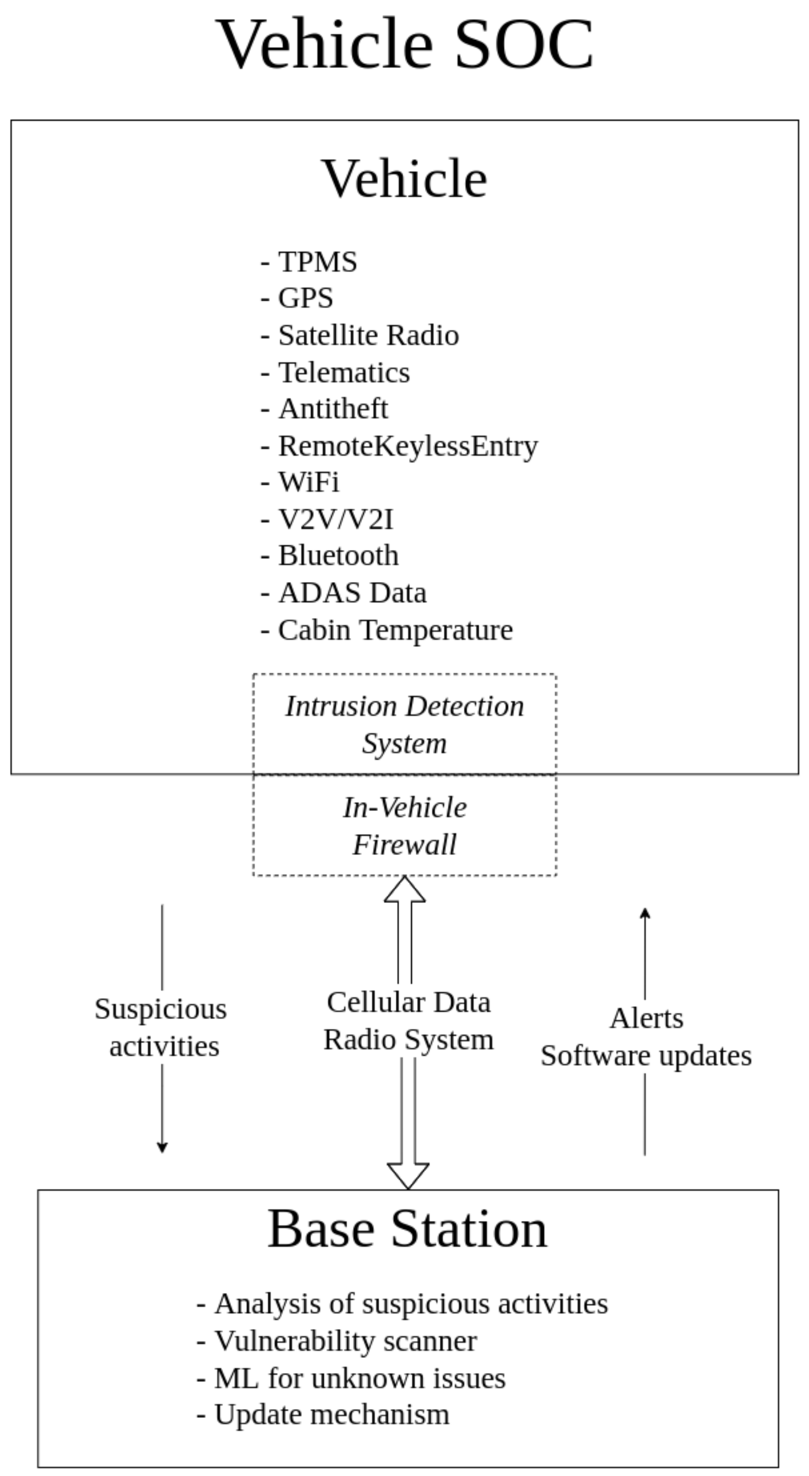

Figure 4.6: High-level schema of VSOC

The main idea of these systems consists in collecting data from various sources, processing it using advanced analysis tools, comparing them with existing attacks, researching for footprints, and actively answering by sending a response to the vehicle as an alert or a software update. A high-level schema is shown in Figure 4.6. The introduction of intrusion detection systems will not be limited to the CAN bus, but every in-vehicle network should be monitored and data collected should be transmitted to base stations.

It is clear that in this situation, intrusion detection systems could help VSOC in obtaining traffic information already processed. Most solutions are closed-source, and there is not much literature on this topic. However, it will become crucial in the near future when new standards and regulations require post-development management and analysis for maintenance. Some problems still need to be solved: firstly, the amount of data transmitted should be limited because all this information must be sent over cellular networks, and vehicles must not saturate the traffic with gigabytes of data not meant for the analysis. Secondly, the transmission will work only when the network is available, and nowadays is not possible to promise this everywhere. An interesting solution to this problem could be using vehicle-to-vehicle communication in order to reach a place where connectivity works and later on send the suspicious data. This solution raises another problem: data transmission must be secured and anonymized. Base stations must receive original data without any modification, and the management of this data should follow privacy regulations enforced in the countries the vehicle is registered. If this information is not used for an immediate response, the storage should be anonymized. Finally, another issue that needs to be considered during the design and implementation of vehicle security operations centers is how the response is treated and what type of response is sent to the vehicle. As stated in previous sections, an alert to stop the car could be a solution, but it is not always possible to stop it, and the driver should not be surprised to receive it(an alert could fear the driver, causing more damage). An advanced solution would be the introduction of software updates that resolve the vulnerability and reacts to the attack. This solution is not simple to implement, especially in systems like the CAN bus, where updates could require modification of protocols or hardware modifications. Moreover, updates should require time and bandwidth, and the vehicle should not be moving during the installation for safety reasons (the update may fail or be corrupted).

5 Conclusions

The study work presented in this thesis tried to achieve multiple goals. Firstly, the analysis, simulation, and comparison of different CAN bus intrusion detection systems against disruptive attacks. The simulation was designed and programmed; data was prepared starting from clean traffic data provided by the University of Turku, where malicious packets were injected. Moreover, an overview of the different vehicle security regulations was brought, with a highlight on ISO/SAE 21434, one of the standards that will have a significant impact on the future of vehicle development.

Based on the literature review and the results obtained during the analysis of CAN bus intrusion detection systems, interesting conclusions can be made.

The analysis, simulation, and comparison of different CAN bus intrusion detection systems were an exciting testing ground and overview of what technology manufacturers can implement on new products to set an effective security valuable layer for monitoring in-vehicle communication channels after the release. The analysis was made on CAN bus traffic of heavy vehicles without prior knowledge of the state of the trucks. The usage of CAN bus technology will still be used in the following years. Based on the fact that the protocol was not designed with security in mind, the importance of monitoring systems will be crucial for complying with the regulations. It is also expected that new protocols may be used in the future in order to add layers of protection that are proactive instead of just reactive.

Results have shown that some intrusions detection systems (frequency-based, and transition-based are effective in detecting denial of service and some types of fuzzing attacks because a high amount of traffic into the CAN bus is injected. On the other hand, targeted attacks were more challenging to detect because the packets tended to be more coherent with regular traffic, and fewer frames were sent through the bus. The most impactful limitation encountered during testing was not knowing in what conditions the data was taken and what was the state of the truck (moving, idle, turning on or off). It is important to remember that the conditions of the vehicle may change the message rate and the state of the CAN bus, so it is necessary to train these intrusion detection systems differently when they are deployed in production.

The importance of introducing security in automotive has been already proven as fundamental: the attack surface area is high and expanding, and malicious actors have already exploited multiple vulnerabilities in the recent past, causing severe disruptions and data leaks. Manufacturers and suppliers can not ignore the problem anymore, and the introduction of additional security and privacy standards allows finding guidance during this mental and priority shift. Regulations are not in place yet, so it is difficult to verify if these documents will radically change the sector's mode of operation. Nevertheless, they are expected to have a significant impact on how new technologies will be implemented and secured. The fact that these regulations will guide the manufacturers during all the life cycles of the vehicle will help introduce security concepts from the design phase until the decommission. The introduction of ISO/SAE 21434 and UNECE 155 will be a turning point both for development and security companies, who will have to adapt significantly to new rules and suggestions to prove the security of the new systems released correctly.

In conclusion, the results can be considered a partial success since interesting information was retrieved by analyzing how the different intrusion detection systems react to different attacks. However, they were not perfect in every condition, meaning that more studies should be made on the development of IDS and the attack patterns. The introduction of automotive security regulations will be a turning point for those studies, but at the same time, attacks will become more advanced and complex to detect. These systems are the correct starting point for monitoring CAN bus traffic and react to potential threats.

References

[1] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle”, *Black Hat USA*, vol. 2015, no. S 91, p. 92, 2015.

[2] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, “Security and privacy vulnerabilities of {in-car} wireless networks: A tire pressure monitoring system case study”, in *19th USENIX Security Symposium (USENIX Security 10)*, 2010.

[3] A. Greenberg, “This Bluetooth Attack Can Steal a Tesla Model X in Minutes”, en-US, *Wired*, Section: tags, ISSN: 1059-1028. [Online]. Available: `https://www.wired.com/story/tesla-model-x-hack-bluetooth/` (visited on 11/20/2021).

[4] International Organization for Standardization, “ISO/SAE 21434 Road vehicles — Cybersecurity engineering”, `https://www.iso.org/standard/70918.html`, 2021.

[5] United Nations Economic and Social Council, “Regulation 155”, `https://unece.org/sites/default/files/2021-03/R155e.pdf`, Mar. 2021.

[6] N. H. T. S. Administration *et al.*, “Cybersecurity best practices for the safety of modern vehicles”, *National Highway Traffic Safety Administration: Washington, DC, USA*, 2020.

[7] "ENISA good practices for security of Smart Cars", en, https://www.enisa.europa.eu/publications/smart-cars, Report/Study. (visited on 12/07/2021).

[8] M. Gmiden, M. H. Gmiden, and H. Trabelsi, "An intrusion detection method for securing in-vehicle CAN bus", in *2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Dec. 2016, pp. 176–180. DOI: 10.1109/STA.2016.7952095.

[9] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1577–1583. DOI: 10.1109/IVS.2017.7995934.

[10] D. Stabili, M. Marchetti, and M. Colajanni, *Detecting attacks to internal vehicle networks through Hamming distance.* Sep. 2017, Pages: 6. DOI: 10.23919/AEIT.2017.8240550.

[11] Forbes, "Supercomputers On Wheels: Creating Safe, Smart Vehicles For Consumers", https://www.forbes.com/sites/forbesbusinesscouncil/2021/10/13/supercomputers-on-wheels-creating-safe-smart-vehicles-for-consumers/, [Online; accessed 2021-11-08], Oct. 2021.

[12] "Number of automotive ECUs continues to rise", https://www.eenewsautomotive.com/news/number-automotive-ecus-continues-rise, [Online; accessed 2021-11-08], May 2019.

[13] "What is OBDII? History of on-board diagnostics", https://www.geotab.com/blog/obd-ii/, [Online; accessed 2021-11-08], Nov. 2020.

[14] International Organization for Standardization, "Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics", https://www.iso.org/standard/64636.html, 2016.

[15] T. Lin and L. Chen, "Common attacks against car infotainment systems", https://events19.linuxfoundation.org/wp-content/uploads/2018/07/ALS19-Common-Attacks-Against-Car-Infotainment-Systems.pdf, [Online; accessed 2022-01-20], 2019.

[16] M. Levi, Y. Allouche, and A. Kontorovich, "Advanced analytics for connected car cybersecurity", in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, IEEE, 2018, pp. 1–7.

[17] S. T. Solutions, "TPMS Legislation Overview in the EU", 2021. [Online]. Available: https://www.schradertpms.com/en-gb/company/press-room/tpms-legislation-overview-eu (visited on 01/03/2022).

[18] G. Dede, R. Hamon, H. Junklewitz, R. Naydenov, A. Malatras, and I. Sanchez, "Cybersecurity challenges in the uptake of artificial intelligence in autonomous driving", *EUR 30568 EN, Publications Office of the European Union*, 2021.

[19] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "Vanet security challenges and solutions: A survey", *Vehicular Communications*, vol. 7, pp. 7–20, 2017.

[20] F. Gallardo and A. P. Yuste, "Scer spoofing attacks on the galileo open service and machine learning techniques for end-user protection", *IEEE Access*, vol. 8, pp. 85 515–85 532, 2020. DOI: 10.1109/ACCESS.2020.2992119.

[21] C. Miller and C. Valasek, "Adventures in automotive networks and control units", *Def Con*, vol. 21, no. 260-264, pp. 15–31, 2013.

[22] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces", in *20th USENIX Security Symposium (USENIX Security 11)*, 2011, p. 16.

[23] L. Christensen and D. Dannberg, "Ethical hacking of iot devices: Obd-ii dongles", 2019.

[24] DEFCONConference, "DEF CON 23 - Samy Kamkar - Drive it like you Hacked it: New Attacks and Tools to Wireles", Dec. 2015. [Online]. Available: `https://www.youtube.com/watch?v=UNgvShN4USU` (visited on 01/03/2022).

[25] "Upstream Security: 2020 Global Automotive Cyber security Report", en, *Network Security*, vol. 2020, no. 1, pp. 4–4, Jan. 2020, ISSN: 1353-4858, 1872-9371. DOI: `10.1016/S1353-4858(20)30005-2`. [Online]. Available: `http://www.magonlinelibrary.com/doi/10.1016/S1353-4858%5C%2820%5C%2930005-2` (visited on 01/03/2022).

[26] "CVE-2018-7800", Available from MITRE, CVE-ID CVE-2018-7800. 2018. [Online]. Available: `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-7800`.

[27] "CVE-2021-22707", Available from MITRE, CVE-ID CVE-2021-22707. 2021. [Online]. Available: `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22707`.

[28] "CVE-2021-22706", Available from MITRE, CVE-ID CVE-2021-22706. 2021. [Online]. Available: `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22706`.

[29] "CVE-2021-22708", Available from MITRE, CVE-ID CVE-2021-22708. 2021. [Online]. Available: `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22708`.

[30] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar", *Black Hat Europe*, vol. 11, no. 2015, p. 995, 2015.

[31] Q. Meng, L.-T. Hsu, B. Xu, X. Luo, and A. El-Mowafy, "A GPS Spoofing Generator Using an Open Sourced Vector Tracking-Based Receiver", en, *Sensors*, vol. 19, no. 18, p. 3993, Jan. 2019, Number: 18 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/s19183993. (visited on 11/28/2021).

[32] International Organization for Standardization, "Road vehicles — Functional safety", https://www.iso.org/standard/68383.html, 2018.

[33] ——, "ISO 31000 Risk Management", https://www.iso.org/iso-31000-risk-management.html, 2018.

[34] M. Security, "The Threats to Our Products", en-US, Aug. 2009. [Online]. Available: https://www.microsoft.com/security/blog/2009/08/27/the-threats-to-our-products/ (visited on 01/11/2022).

[35] Trend Micro, "ISO/SAE 21434 - Setting the Standard for Connected Cars' Cybersecurity", https://resources.trendmicro.com/WP-Connected-Cars.html, Jun. 2020.

[36] United Nations Economic and Social Council, "ECE/TRANS/WP.29/2020/79", https://unece.org/fileadmin/DAM/trans/doc/2020/wp29grva/ECE-TRANS-WP29-2020-079-Revised.pdf, Jun. 2020.

[37] Auto ISAC, "Best Practices - Auto-ISAC", en-US. [Online]. Available: https://automotiveisac.com/best-practices/ (visited on 01/11/2022).

[38] "FTC And NHTSA Navigate Privacy And Security Issues At Connected Cars Workshop". [Online]. Available: https://www.mondaq.com/unitedstates/security/608210/ftc-and-nhtsa-navigate-privacy-and-security-issues-at-connected-cars-workshop (visited on 12/07/2021).

[39] "General Data Protection Regulation (GDPR) - Official Legal Text", en-US, 2016. [Online]. Available: https://gdpr-info.eu/ (visited on 01/11/2022).

[40] "Guidelines 1/2020 on processing personal data in the context of connected vehicles and mobility related applications", 2020. [Online]. Available: https://edpb.europa.eu/our-work-tools/documents/public-consultations/2020/guidelines-12020-processing-personal-data_en.

[41] CiA, "CAN in Automation (CiA): History of the CAN technology". [Online]. Available: https://www.can-cia.org/can-knowledge/can/can-history/ (visited on 12/11/2021).

[42] C. Specification, "Bosch", *Robert Bosch GmbH, Postfach*, vol. 50, 1991.

[43] CSS Electronics, "CAN Bus Explained - A Simple Intro [v2.0 | 2021]", Oct. 2021. [Online]. Available: https://www.youtube.com/watch?v=oYps7vT708E (visited on 12/11/2021).

[44] International Organization for Standardization, "Road vehicles — Controller area network (CAN)", https://www.iso.org/standard/63648.html, 2015.

[45] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of CAN Bus Security Challenges", en, *Sensors*, vol. 20, no. 8, p. 2364, Jan. 2020, Number: 8 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/s20082364. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2364 (visited on 12/13/2021).

[46] S. Hartzell and C. Stubel, "Automobile can bus network security and vulnerabilities", 2018. [Online]. Available: https://canvas.uw.edu/files/47669787/download?download_frd=1 (visited on 12/02/2021).

[47] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection", in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 911–927.

[48] R. Kammerer, B. Framel, and A. Wasicek, “Enhancing security in CAN systems using a star coupling router”, in *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*, ISSN: 2150-3117, Jun. 2012, pp. 237–246. DOI: 10.1109/SIES.2012.6356590.

[49] R. Elder, C. Westrick, and P. Moldenhauer, “Cyberattack detection and bus segmentation in ground vehicles”, in *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium*, 2020, pp. 11–13.

[50] Z. Lu, Q. Wang, X. Chen, G. Qu, Y. Lyu, and Z. Liu, “Leap: A lightweight encryption and authentication protocol for in-vehicle communications”, in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 1158–1164.

[51] W. A. Farag, “CANTrack: Enhancing automotive CAN bus security using intuitive encryption algorithms”, in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, Apr. 2017, pp. 1–5. DOI: 10.1109/ICMSAO.2017.7934878.

[52] M. Jukl and J. ÄŒupera, “Using of tiny encryption algorithm in CAN-Bus communication”, en, *Research in Agricultural Engineering*, vol. 62, no. No. 2, pp. 50–55, Jun. 2016, ISSN: 12129151, 18059376. DOI: 10.17221/12/2015-RAE. [Online]. Available: http://www.agriculturejournals.cz/web/rae.htm?volume=62&firstPage=50&type=publishedArticle (visited on 12/24/2021).

[53] S. J. Shepherd, “The Tiny Encryption Algorithm”, en, *Cryptologia*, vol. 31, no. 3, pp. 233–245, Jul. 2007, ISSN: 0161-1194, 1558-1586. DOI: 10.1080/01611190601090606. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01611190601090606 (visited on 12/24/2021).

[54] C. H. Park, Y. Kim, and J.-Y. Jo, “A Secure Communication Method for CANBus”, in *2021 IEEE 11th Annual Computing and Communication Work-*

shop and Conference (CCWC), Jan. 2021, pp. 0773–0778. DOI: 10.1109/CCWC51732.2021.9376166.

[55] T. P. Doan and S. Ganesan, "CAN Crypto FPGA Chip to Secure Data Transmitted Through CAN FD Bus Using AES-128 and SHA-1 Algorithms with A Symmetric Key", English, SAE International, Warrendale, PA, SAE Technical Paper 2017-01-1612, Mar. 2017, ISSN: 0148-7191, 2688-3627. DOI: 10.4271/2017-01-1612. [Online]. Available: https://www.sae.org/publications/technical-papers/content/2017-01-1612/ (visited on 12/26/2021).

[56] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels", in *Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000*, ISSN: 1081-6011, May 2000, pp. 56–73. DOI: 10.1109/SECPRI.2000.848446.

[57] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks", en, *Wireless Networks*, vol. 8, no. 5, pp. 521–534, Sep. 2002, ISSN: 1572-8196. DOI: 10.1023/A:1016598314198. [Online]. Available: https://doi.org/10.1023/A:1016598314198 (visited on 12/26/2021).

[58] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus", in *ECRYPT workshop on Lightweight Cryptography*, ECRYPT, vol. 2011, 2011, p. 20.

[59] T. Ziermann, S. Wildermann, and J. Teich, *CAN+: A new backward-compatible Controller Area Network (CAN) protocol with up to 16Ã— higher data rates.* May 2009, Journal Abbreviation: Proceedings -Design, Automation and Test in Europe, DATE Pages: 1093 Publication Title: Proceedings -Design, Automation and Test in Europe, DATE. DOI: 10.1109/DATE.2009.5090826.

[60] S.-F. Lokman, A. T. Othman, and M.-H. Abu-Bakar, "Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review", en, *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 184, Dec. 2019, ISSN: 1687-1499. DOI: 10.1186/s13638-019-1484-3. [Online]. Available: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-019-1484-3 (visited on 12/28/2021).

[61] M. Kneib and C. Huth, "Scission: Signal Characteristic-Based Sender Identification and Intrusion Detection in Automotive Networks", in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18, New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 787–800, ISBN: 978-1-4503-5693-0. DOI: 10.1145/3243734.3243751. [Online]. Available: https://doi.org/10.1145/3243734.3243751 (visited on 12/29/2021).

[62] I. Studnia, E. Alata, V. Nicomette, M. Kaâniche, and Y. Laarouchi, "A language-based intrusion detection approach for automotive embedded networks", *International Journal of Embedded Systems*, vol. 10, no. 1, 2018.

[63] S. Jin, J.-G. Chung, and Y. Xu, "Signature-Based Intrusion Detection System (IDS) for In-Vehicle CAN Bus Network", in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, ISSN: 2158-1525, May 2021, pp. 1–5. DOI: 10.1109/ISCAS51556.2021.9401087.

[64] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks", in *2008 IEEE Intelligent Vehicles Symposium*, ISSN: 1931-0587, Jun. 2008, pp. 220–225. DOI: 10.1109/IVS.2008.4621263.

[65] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN)

Timing", *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1484–1494, Feb. 2020, Conference Name: IEEE Transactions on Vehicular Technology, ISSN: 1939-9359. DOI: `10.1109/TVT.2019.2961344`.

[66] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network", in *2016 International Conference on Information Networking (ICOIN)*, Jan. 2016, pp. 63–68. DOI: `10.1109/ICOIN.2016.7427089`.

[67] C. Ling and D. Feng, "An Algorithm for Detection of Malicious Messages on CAN Buses", en, ISSN: 1951-6851, Atlantis Press, Nov. 2012, pp. 627–630, ISBN: 978-94-91216-39-8. DOI: `10.2991/citcs.2012.161`. [Online]. Available: `https://www.atlantis-press.com/proceedings/citcs-12/3107` (visited on 09/16/2021).

[68] T. Hoppe, S. Kiltz, and J. Dittmann, "Applying intrusion detection to automotive IT-early insights and remaining challenges", *Journal of Information Assurance and Security (JIAS)*, vol. 4, pp. 226–235, Jan. 2009.

[69] M.-J. Kang and J.-W. Kang, "A Novel Intrusion Detection Method Using Deep Neural Network for In-Vehicle Network Security", in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–5. DOI: `10.1109/VTCSpring.2016.7504089`.

[70] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based Intrusion Detection System for In-Vehicle Network", in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, Aug. 2018, pp. 1–6. DOI: `10.1109/PST.2018.8514157`.

[71] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks", in

2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Oct. 2016, pp. 130–139. DOI: `10.1109/DSAA.2016.20`.

[72] M. Lombardi, F. Pascale, and D. Santaniello, “Eids: Embedded intrusion detection system using machine learning to detect attack over the can-bus”, in *Proceedings of the 30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference, Venice, Italy*, 2020, pp. 21–26.

[73] R. E. Haas, D. P. F. Maller, P. Bansal, R. Ghosh, and S. S. Bhat, “Intrusion detection in connected cars”, in *2017 IEEE International Conference on Electro Information Technology (EIT)*, ISSN: 2154-0373, May 2017, pp. 516–519. DOI: `10.1109/EIT.2017.8053416`.

[74] R. Islam, M. K. Devnath, M. D. Samad, and S. M. J. A. Kadry, “GGNB: Graph-based Gaussian naive Bayes intrusion detection system for CAN bus”, *Vehicular Communications*, vol. 33, p. 100 442, 2022, ISSN: 2214-2096. DOI: `https://doi.org/10.1016/j.vehcom.2021.100442`.

[75] Q. Wang, Z. Lu, and G. Qu, “An entropy analysis based intrusion detection system for controller area network in vehicles”, in *2018 31st IEEE International System-on-Chip Conference (SOCC)*, IEEE, 2018, pp. 90–95.

[76] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, “Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms”, in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, Sep. 2016, pp. 1–6. DOI: `10.1109/RTSI.2016.7740627`.

[77] M. Muter and N. Asaj, “Entropy-based anomaly detection for in-vehicle networks”, in *2011 IEEE Intelligent Vehicles Symposium (IV)*, ISSN: 1931-0587, Jun. 2011, pp. 1110–1115. DOI: `10.1109/IVS.2011.5940552`.

[78] H. Lee, S. H. Jeong, and H. K. Kim, “OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame”, in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, Aug. 2017, pp. 57–5709. DOI: `10.1109/PST.2017.00017`.

[79] M. Weber, S. Klug, E. Sax, and B. Zimmer, “Embedded Hybrid Anomaly Detection for Automotive CAN Communication”, in *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*, Toulouse, France, Jan. 2018.

[80] D. Grimm, M. Weber, and E. Sax, “An Extended Hybrid Anomaly Detection System for Automotive Electronic Control Units Communicating via Ethernet - Efficient and Effective Analysis using a Specification- and Machine Learning-based Approach”, in *VEHITS*, 2018. DOI: `10.5220/0006779204620473`.

[81] D. M. Kang, S. H. Yoon, D. K. Shin, Y. Yoon, H. M. Kim, and S. H. Jang, “A Study on Attack Pattern Generation and Hybrid MR-IDS for In-Vehicle Network”, in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Apr. 2021, pp. 291–294. DOI: `10.1109/ICAIIC51459.2021.9415261`.

[82] hardbyte, “python-can”, `https://python-can.readthedocs.io/en/master`, [Online; accessed 2022-01-20], 2022.

[83] SAE International, “Physical layer, 250 kbps, twisted shielded pair”, `https://www.sae.org/standards/content/j1939/11_201612/`, [Online; accessed 2022-01-20], 2016.

[84] ——, “Data link layer”, `https://www.sae.org/standards/content/j1939/21_201810/`, [Online; accessed 2022-01-20], 2018.

[85] ——, “Network layer”, `https://www.sae.org/standards/content/j1939/31_201809/`, [Online; accessed 2022-01-20], 2018.